IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of	:	**Confirmation No. 8029**
Tomoyuki OKADA et al.	:	Docket No. PR02_1189A
Serial No. 60/409,999	:	
Filed September 12, 2002	:	

APPARATUS, METHOD, AND SYSTEM
FOR REPRODUCING IMAGE CONTENTS
RECORDING MEDIUM

SUBMISSION OF ENGLISH TRANSLATION WITH STATEMENT OF ACCURACY

Assistant Commissioner for Patents,
Washington, D.C.

Sir:

The above-identified U.S. provisional patent application was filed on September 12, 2002 in a language other than English. Therefore, in accordance with 37 CFR 1.78(a)(5)(iv), an English language translation of the provisional application is submitted herewith together with a statement that the translation is accurate.

Respectfully submitted,

Tomoyuki OKADA et al.

By:_____
Charles R. Watts
Registration No. 33,142
Attorney for Applicants

CRW/asd
Washington, D.C. 20006-1021
Telephone (202) 721-8200
Facsimile (202) 721-8250
November 21, 2002

Exhibit 3

VERIFICATION OF TRANSLATION

I, Sachiko Takagi, translator of 1-18-16, Senriyamanishi, Suita, Osaka, Japan, hereby declare that I am conversant with the English and Japanese languages and am a competent translator thereof. I further declare that to the best of my knowledge and belief the following is a true and correct translation made by me of U.S. Provisional Application No. 60/409999 filed on September 12, 2002.

Date: November 8, 2002

Sachiko Takagi

SACHIKO TAKAGI

TITLE OF THE INVENTION

APPARATUS, METHOD, AND SYSTEM FOR REPRODUCING IMAGE CONTENTS RECORDING MEDIUM

BACKGROUND OF THE INVENTION

(1) Field of the Invention

The present invention relates to an apparatus, method, and system for playing back a medium on which digital image content is recorded.

(2) Description of the Related Art

The following describes a conventional DVD (hereafter referred to as "SD-DVD" or simply "DVD") technology.

FIG. 1 shows the structure of a SD-DVD. As shown in the lower part of FIG. 1, the DVD disc has a logical address space between the lead-in and the lead-out. Volume information of a file system is recorded at the start of the logical address space. Application data such as video and audio is recorded following the volume information.

A file system is a system for describing data written on a disc in units of directories and files. Example file systems include ISO 9660 and UDF (Universal Disc Format). A PC (personal computer) of daily use can

present data stored in a hard disk in units of directories and files, through a file system such as FAT (File Allocation Table) or NTFS (New Technology File System). This enhances usability.

SD-DVDs use both UDF and ISO 9660 (a combination of the two is sometimes called "UDF Bridge"). Accordingly, data can be read from a SD-DVD using a file system driver of any of UDF and ISO 9660 (the DVD referred to here is a ROM disc for package-media and therefore cannot be physically written).

Data recorded on a DVD can be seen as directories or files shown in the upper left part of FIG. 1, through the use of UDF Bridge. A directory called VIDEO_TS is provided immediately below a ROOT directory. Application data of the DVD is stored in this VIDEO_TS directory. The application data is stored as a plurality of files. Main files include:

VIDEO_TS.IFO (disc playback control information file)

VTS_01_0.IFO (video title set#1 playback control information file)

VTS_01_0.VOB (video title set#1 stream file)

.....

There are two types of extensions. "IFO" indicates that the file contains playback control information, whereas "VOB" indicates that the file contains an MPEG stream that is AV data. Playback control information includes information for providing interactivity adopted for DVDs (the ability of dynamically changing playback (reproduction) in response to user operations), and information such as metadata that accompanies titles or AV streams. In DVDs, playback control information is sometimes called navigation information.

Of the above listed files, VIDEO_TS.IFO and VTS_01_0.IFO are playback control information files. VIDEO_TS.IFO manages the overall disc. VTS_01_0.IFO is playback control information for an individual video title set (note that a plurality of titles, such as different movies or different versions of one movie, can be recorded on a single DVD). In VTS_01_0.IFO, "01" in the body of the file name designates the number of the video title set. For instance, VTS_02_0.IFO is a playback control information file for the video title set#2.

The upper right part of FIG. 1 shows a DVD

navigation space in the application layer of the DVD. This is a logical structure space where the above playback control information has been expanded. In the DVD navigation space, the information in the VIDEO_TS.IFO file has been expanded as VMGI (Video Manager Information), whilst the playback control information in each of the VTS_01_0.IFO file and the files for the other video title sets has been expanded as VTSI (Video Title Set Information).

The VTSI includes PGCI (Program Chain Information). The PGCI is information for a playback sequence called a PGC (Program Chain). The PGCI is made up of a collection of cells and a kind of programming information called commands. Each cell constitutes all or part of a VOB (Video Object, representing an MPEG stream). Playing a cell is an equivalent of playing part of a VOB that is specified by the cell.

Commands are processed by a DVD virtual machine, and are similar to JavaScript or the like executed on a browser. However, DVD commands differ from JavaScript in the following point. JavaScript controls windows and browsers (e.g. open a window of a new browser), in addition to performing logical operations. In contrast, DVD commands only control playback of AV titles, such as by designating a chapter to be played, in addition

to performing logical operations.

A cell contains the start and end addresses (logical addresses) of a VOB recorded on the disc, as its internal information. A player reads data using the start and end addresses of the VOB written in the cell, and plays back the data.

FIG. 1 is a schematic diagram for explaining navigation information embedded in an AV stream. The interactivity featured by SD-DVDs is achieved not only by the navigation information stored in files such as VIDEO_TS.IFO and VTS_01_0.IFO. Other important information has been multiplexed with video and audio data in VOBs, using dedicated carriers called navigation packs (navi-packs or NV_PCKs).

The following explains a menu, as a simple example of interactivity. A menu has several buttons. For each button, a process to be executed when that button is selected has been defined. One button is currently selected on the menu (the selected button is highlighted by overlaying a translucent highlight color). The user can move the highlight to other buttons positioned up, down, left, and right, using the Up/Down/Left/Right keys on a remote controller. When the user moves the highlight to a button he or she wants to select using the Up/Down/Left/Right keys on the remote controller and

confirms the selection (by pressing the Return key), a program of a command corresponding to the selected button is executed. In general, playback of a title or a chapter is executed by a command.

The upper left part of FIG. 2 gives a brief description of an NV_PCK.

This NV-PCK contains highlight color information and button information for each button. The highlight color information includes color palette information, and specifies a translucent highlight color used for overlaying. The button information of each button includes rectangular area information, move information, and button command information. The rectangular area information shows the position of the button. The move information shows movements of the button to other buttons (designation of buttons to which the highlight is to be moved in response to the user's operations of the Up/Down/Left/Right keys). The button command information shows a command to be executed when the selection of the button is confirmed.

The highlight on the menu is shown as an overlay image, as shown in the central upper right part of FIG. 2. The overlay image is formed by painting the color specified by the color palette information on a rectangular area shown by the rectangular area

information. This overlay image is combined with a background image shown in the right part of FIG. 2, and the resulting image is displayed on the screen.

Menus in DVDs are realized in this way. The reason that navigation data is partly embedded in a stream using NV_PCKs is to enable menu information to be dynamically updated in sync with the stream. In so doing, even if an application has difficult synchronization timing, such as when displaying a menu only for five to ten minutes during playback of a movie, the application can be realized without problem.

FIG. 3 shows a VOB on a DVD. As illustrated, data such as video, audio, and subtitles (FIG. 3A) is divided into packets and packs based on the MPEG system (ISO/IEC 13818-1) standard (FIG. 3B). These are then multiplexed into one MPEG program stream (FIG. 3C). NV_PCKs containing button commands for realizing interactivity are also multiplexed in this MPEG program stream, as described above.

Multiplexing in MPEG has the following characteristics. A bit sequence within each individual packet which is multiplexed in an MPEG program stream is formed according to the decoding order of data in the packet. However, adjacent packets multiplexed in the MPEG program stream, i.e., video, audio, and subtitle

data, are not necessarily arranged according to their playback or decoding order. This is because an MPEG system stream decoder model (generally called a "System Target Decoder" or "STD" (FIG. 3D)) has decoder buffers corresponding to individual elementary streams, and stores demultiplexed data in the corresponding decoder buffers until decoding. These decoder buffers have different sizes. In more detail, a video buffer has a capacity of 232kB, an audio buffer has a capacity of 4kB, and a subtitle buffer has a capacity of 52kB. Since a data input timing differs for each decoder buffer, there is a difference between the order in which bit sequences of packets are arranged in an MPEG system stream and the order in which these bit sequences are decoded.

Which is to say, even if subtitle data is multiplexed adjacent to video data, it does not necessarily mean that the subtitle data will be decoded at the same time as the video data.

(Problems the Invention is Attempting to Solve)

Due to recent advances in IT technology, movie viewing is no longer restricted to a style where a viewer connects a DVD to a TV and plays back the DVD. Nowadays, people can watch movies on PCs, or even connect DVD players to the Internet like PCs. As a result, the

conventional division between DVDs as AV devices and PCs as non-AV devices is breaking down. In the future, new types of entertainment that combine AV content recorded on DVDs with the Internet are expected to increase.

In view of this, the problem of SD-DVDs is that information which offers interactivity is embedded in a stream using NV_PCKs, as explained with reference to FIG. 2.

One advantage of using the Internet is the ability of downloading new content from a server, as represented by FTP (File Transport Protocol). This delivers new types of enjoyment which are impossible with the conventional DVD technology. For example, new data such as video, audio, and subtitles can be downloaded from the Internet. Also, such new data can be played in sync with data recorded on a disc, by streaming. Furthermore, playback control information, i.e., an interactive scenario, can be updated to enjoy a movie with a new story and new ending.

SUMMARY OF THE INVENTION

In view of the above problem, the present invention aims to provide a HD-DVD technology that realizes new applications where disc media are allied with the Internet, which are impossible with conventional DVDs.

The stated object can be achieved by a playback apparatus for playing back data on an optical disc on which at least video data and audio data are recorded, including: a plurality of application playback modules; and a controller for selecting an application playback module from the plurality of application playback modules.

Here, the playback apparatus includes a basic playback unit for performing a basic playback operation that is commonly used by the plurality of application playback modules.

Here, the basic playback unit includes: a status storage unit for storing information showing a status of the playback apparatus; and a playback control unit for controlling playback of the video data and the audio data.

The stated object can also be achieved by a playback system including: an optical disc on which at least video data and audio data are recorded; and a playback apparatus for playing back data on the optical disc, the playback apparatus including: a plurality of application playback modules; and a controller for automatically selecting, if an application module is stored in a predetermined directory on the optical disc, a predetermined application playback module from the

plurality of application playback modules, wherein the application module is activated by the selected application playback module.

The stated object can also be achieved by a playback method for playing back data on an optical disc on which at least video data and audio data are recorded, including: a plurality of application playback steps; and a control step for selecting an application playback step from the plurality of application playback steps.

BRIEF DESCRIPTION OF THE DRAWINGS

These and other objects, advantages and features of the invention will become apparent from the following description thereof taken in conjunction with the accompanying drawings which illustrate a specific embodiment of the invention.

In the drawings:

FIG. 1 shows the construction of a DVD;

FIG. 2 shows the structure of highlighting;

FIG. 3 shows an example of multiplexing on a DVD;

FIG. 4 shows the data hierarchy of an HD-DVD;

FIG. 5 shows the structure of a logical space on the HD-DVD;

FIG. 6 is a schematic block diagram of an HD-DVD player;

FIG. 7 is a block diagram showing the construction of the HD-DVD player;

FIG. 8 is a diagram for explaining an application space in the HD-DVD;

FIG. 9 shows the structure of an MPEG stream (VOB);

FIG. 10 shows the structure of a pack;

FIG. 11 is a diagram for explaining the relationship between an AV stream and the player construction;

FIG. 12 shows a model for continuously supplying AV data to a track buffer;

FIG. 13 shows the structure of a VOB information file;

FIG. 14 is a diagram for explaining a time map;

FIG. 15 is a diagram for explaining a method of acquiring address information using the time map;

FIG. 16 shows the structure of a playlist file;

FIG. 17 shows the structure of a program file corresponding to a playlist;

FIG. 18 shows the structure of a BD disc overall management information file;

FIG. 19 shows the structure of a file storing global event handlers;

FIG. 20 is a diagram for explaining an example of a time event;

FIG. 21 is a diagram for explaining an example of a user event;

FIG. 22 is a diagram for explaining a global event handler;

FIG. 23 shows the structure of a virtual machine;

FIG. 24 shows a player variable table;

FIG. 25 shows an example of event handlers (for time events);

FIG. 26 shows an example of an event handler (for a user event);

FIG. 27 is a flowchart showing basic processing performed by the player;

FIG. 28 is a flowchart showing playlist playback processing;

FIG. 29 is a flowchart showing event processing;

FIG. 30 is a flowchart showing subtitle processing;

FIG. 31 is a diagram for explaining a concept of BD applications;

FIG. 32 shows the logical structure of a BD in the second embodiment;

FIG. 33 is a diagram for explaining a BD basic playback feature in the second embodiment;

FIG. 34 shows the directory/file structure of the BD in the second embodiment;

FIG. 35 is a block diagram showing the construction

of a BD player in the second embodiment;

FIG. 36 shows the structure of a virtual machine of the BD player in the second embodiment;

FIG. 37 is a diagram for explaining an example of execution in a Java class;

FIG. 38 is a flowchart showing basic processing performed by the BD player in the second embodiment;

FIG. 39 is a flowchart showing event processing in the second embodiment; and

FIG. 40 is a flowchart showing switching between feature modules by the BD player.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

(First Embodiment)

The following is a description of the first embodiment of the present invention.

(Logical Data Structure on a Disc)

FIG. 4 shows the structure of an HD-DVD (hereafter also referred to as a "BD"). In particular, FIG. 4 shows the structure of a BD disc (104) which is a disc medium, and the structure of data (101, 102, and 103) recorded on the disc. The data recorded on the BD disc (104) includes AV data (103), BD management information (102) including management information relating to the AV data

and AV playback sequences, and BD playback programs (101) for realizing interactivity. This embodiment describes the case where the BD disc is mainly used in AV applications for reproducing AV content such as movies. However, the BD disc may of course be used as a recording medium for use in computers, like a CD-ROM or a DVD-ROM.

FIG. 5 shows logical data recorded on the above BD disc. Like other optical discs such as DVDs and CDs, the BD disc has a spiral recording area that runs from its inner radius to outer radius, and has a logical address space for storing logical data between the lead-in at the inner radius and the lead-out at the outer radius. Also, a special area called a BCA (Burst Cutting Area) which can only be read by a drive is provided on the inner side of the lead-in. This area cannot be read by an application, and so often aids in copyright protection and similar techniques.

File system information (volume) is stored at the start of the logical address space. Application data such as video data is stored following the file system information. Example file systems include UDF and ISO 9660, as explained in the Description of the Related Art. A file system enables logical data which is stored in the same manner as in usual PCs, to be read through a

directory/file structure.

The BD disc of this embodiment has the following directory/file structure. A BDVIDEO directory is provided immediately below a ROOT directory. The BDVIDEO directory stores data (101, 102, and 103 in FIG. 4) such as the AV content and management information used in the HD-DVD.

The following seven types of files are provided under the BDVIDEO directory.

BD.INFO (the file name is fixed)
Classified as BD management information, and store information relating to the overall BD disc. The BD player reads this file first.

BD.PROG (the file name is fixed)
Classified as a BD playback program, and store a program relating to the overall BD disc.

XXX.PL (XXX is variable while the extension PL is fixed)
Classified as BD management information, and store playlist information containing a scenario. This file is provided for each

playlist.

XXX.PROG (XXX is variable while the extension PROG is fixed)
Classified as a BD playback program, and store a program provided for a playlist. The body (XXX) of the file name identifies the playlist.

YYY.VOB (YYY is variable while the extension VOB is fixed)
Classified as AV data, and store a VOB (same as that explained in the Description of the Related Art). This file is provided for each VOB.

YYY.VOBI (YYY is variable while the extension VOBI is fixed)
Classified as BD management information, and store management information relating to a VOB that is AV data. The body (YYY) of the file name identifies the corresponding VOB.

ZZZ.PNG (ZZZ is variable while the extension PNG is fixed)

Classified as AV data, and store PNG image data for forming a subtitle or a menu (PNG is a graphics format approved as a standard by W3C, and pronounced "ping"). This file is provided for each PNG image.

(Construction of a Player)

The construction of a player for playing back the above BD disc is described below, by referring to FIGS. 6 and 7.

FIG. 6 is a block diagram showing the rough functional construction of the player.

Data on a BD disc (201) is read via an optical pickup (202). The read data is stored in a special memory according to data type. A BD playback program (the contents of the BD.PROG or XXX.PROG file) is stored in a program storage memory (203). BD management information (the contents of the BD.INFO, XXX.PL, or YYY.VOBI file) is stored in a management information storage memory (204). AV data (the contents of the YYY.VOB or ZZZ.PNG file) is stored in an AV storage memory (205).

The BD playback program stored in the program storage memory (203) is processed by a program processing unit (206). The BD management information stored in the

management information storage memory (204) is processed by a management information processing unit (207). The AV data stored in the AV storage memory (205) is processed by a presentation processing unit (208).

The program processing unit (206) receives information, such as information about a playlist to be played and event information showing an execution timing of a program, from the management information processing unit (207). The program processing unit (206) executes a program based on the received information. Here, the program can dynamically change the playback from one playlist to another. This is achieved by the program processing unit (206) sending an instruction to play back the other playlist to the management information processing unit (207). The program processing unit (206) also receives an event generated by the user, i.e., a request made by the user through remote control keys. If there is a program corresponding to the user event, the program processing unit (206) executes that program.

The management information processing unit (207) receives an instruction from the program processing unit (206), and analyzes management information of a playlist corresponding to the instruction and management information of a VOB corresponding to the playlist. The management information processing unit (207) then

instructs the presentation processing unit (208) to play corresponding AV data. Also, the management information processing unit (207) receives reference time information from the presentation processing unit (208), and instructs the presentation processing unit (208) to stop playing the AV data based on the time information. Also, the management information processing unit (207) generates an event showing an execution timing of a program, and sends it to the program processing unit (206).

The presentation processing unit (208) has decoders in a one-to-one correspondence with video data, audio data, and subtitle/image data. The presentation processing unit (208) decodes AV data and outputs the result, according to an instruction from the management information processing unit (207). When video data and subtitle/image data are decoded, they are drawn on respective special planes, namely, a video plane (210) and an image plane (209). A superimposing unit 211 superimposes them together, and outputs the result to a display device of a TV or the like.

Note that the construction of the BD player shown in FIG. 6 is based on the data structure of the BD disc shown in FIG. 4.

FIG. 7 is a block diagram showing the above player

construction in greater detail. In the drawing, the AV storage memory (205) corresponds to and is divided into an image memory (308) and a track buffer (309). The program processing unit (206) corresponds to and is divided into a program processor (302) and a UOP manager (303). The management information processing unit (207) corresponds to and is divided into a scenario processor (305) and a presentation controller (306). The presentation processing unit (208) corresponds to and is divided into a clock (307), a demultiplexer (310), an image processor (311), a video processor (312), and an audio processor (313).

VOB data (MPEG stream) read from the BD disc (201) is stored in the track buffer (309). Image data (PNG) read from the BD disc (201) is stored in the image memory (308). The demultiplexer (310) extracts the VOB data stored in the track buffer (309) based on the time shown by the clock (307), and outputs video data and audio data respectively to the video processor (312) and the audio processor (313). The video processor (312) and the audio processor (313) are each made up of a decoder buffer and a decoder, in accordance with the MPEG system standard. This being so, each of the video and audio data output from the demultiplexer (310) is temporarily stored in the corresponding decoder buffer, and decoded

by the corresponding decoder in accordance with the clock (307).

Meanwhile, the PNG image data stored in the image memory (308) is processed by one of the following two methods.

When the image data is for a subtitle, its decode timing is indicated by the presentation controller (306). First, the scenario processor (305) receives time information from the clock (307). When a subtitle display time (start or end time) is reached, the scenario processor (305) instructs the presentation controller (306) to start or stop the display of the subtitle, so that the display of the subtitle can be appropriately carried out. The presentation controller (306) sends a decode/display instruction to the image processor (311). The image processor (311) accordingly extracts the corresponding image data from the image memory (308), decodes the image data, and draws it on an image plane (314).

When the image data is for a menu, on the other hand, its decode timing is indicated by the program processor (302). The time when the program processor (302) instructs the decoding depends on a BD program which is being executed by the program processor (302), and so cannot be generalized.

As explained with reference to FIG. 6, the image data and video data which have been decoded are drawn on the image plane (314) and video plane (315) respectively. The superimposing unit (316) superimposes them together, and outputs the result.

Management information (scenario or AV management information) read from the BD disc (201) is stored in the management information storage memory (304). The scenario information (BD.INFO or XXX.PL) is read and processed by the scenario processor (305). The AV management information (YYY.VOBI) is read and processed by the presentation controller (306).

The scenario processor (305) analyzes information about a playlist. The scenario processor (305) indicates a VOB referred to by the playlist and a playback position of the VOB, to the presentation controller (306). The presentation controller (306) analyzes management information (YYY.VOBI) of the VOB, and instructs the drive controller (317) to read the VOB.

The drive controller (317) moves the optical pickup and reads corresponding AV data, in accordance with the instruction from the presentation controller (306). The read AV data is stored in the image memory (308) or in the track buffer (309), as described earlier.

Also, the scenario processor (305) monitors the

time shown by the clock (307), and sends an event to the program processor (302) at a timing set by the management information.

A BD program (BD.PROG or XXX.PROG) stored in the program storage memory (301) is executed by the program processor (302). Execution of a BD program by the program processor (302) is launched when an event is sent from the scenario processor (305) or when an event is sent from the UOP manager (303). The UOP manager (303) generates an event to be sent to the program processor (302), upon receiving a request made by the user through remote control keys.

(Application Space)

FIG. 8 shows an application space of the HD-DVD.

In the application space of the HD-DVD, a playlist (Playlist) is a unit of playback. The playlist has a static scenario formed by a playback sequence of cells (Cells), and a dynamic scenario described by a program. If there is no program describing a dynamic scenario, the playback of the playlist is performed by merely playing the individual cells in sequence. In this case, the playback of the playlist is completed once all cells have been played. In contrast, a program can describe a playback sequence other than that of the static

scenario, or dynamically change the target of playback according to the user's selection or the status of the player. A typical example of this is a menu. In HD-DVDs, a menu allows the user to dynamically select a scenario, i.e., a playlist, which is to be played.

The program referred to here is an even handler that is executed according to a time event or a user event.

A time event is an event which is generated based on time information embedded in a playlist. The event sent from the scenario processor (305) to the program processor (302), which has been explained with reference to FIG. 7, is a time event. Once a time event has been issued, the program processor (302) executes an event handler associated with the time event by an ID. As described above, the program executed by the program processor (302) may designate playback of another playlist. In such a case, the playback of the currently-played playlist stops, and instead the playback of the designated playlist begins.

A user event is an event which is generated by the user's remote control key operation. There are roughly two types of user events. The first type is a menu selection event generated by operating the cursor keys (the Up/Down/Left/Right keys) or the Return key. An event handler corresponding to a menu selection event

is valid only in a limited period within a playlist (a valid period for each individual event handler has been specified in information about the playlist). When the user presses any of the Up/Down/Left/Right keys or the Return key on the remote controller, a valid event handler corresponding to the user operation is searched. If the valid event handler is available, that event handler is executed. Otherwise, the menu selection event is ignored.

The second type is a menu call event generated by operating the Menu key. When a menu call event occurs, a global event handler is called. The global event handler is an event handler which does not depend on playlists and is always valid. With the provision of this feature, the menu call function of DVDs (call an audio menu, a subtitle menu, or the like during playback of a title, and resume the playback of the title at a suspended point after the audio or subtitle has been changed).

A cell (Cell) that is a constituent unit of a static scenario of a playlist refers to all or part of a VOB (MPEG stream). The cell has information showing a start time and end time of a playback section of the VOB. VOB management information (VOBI) corresponding to each VOB contains a time map (Time Map or TM). With reference

to this time map, a start address and end address of data to be read from the VOB (i.e. the corresponding YYY.VOB file) can be obtained from the above start and end time of the playback section of the VOB. The time map is explained in greater detail later.

(Details on a VOB)

FIG. 9 shows the structure of an MPEG stream (VOB) used in this embodiment.

As shown in the drawing, a VOB is made up of a plurality of VOBUs (Video Object Units). A VOBU is based on a GOP (Group Of Pictures) defined for MPEG video streams, and is a playback unit as a multiplexed stream including audio data. Each VOBU has a time period of 0.4 to 1.0 second. Usually, each VOBU has a playback time period of 0.5 second. This is because the GOP structure under MPEG is typically 15 frames per second (in the case of NTSC).

A VOBU includes video packs (V_PCKs) and audio packs (A_PCKs). Each pack has one sector, which is 2kB in this embodiment.

FIG. 10 shows the structure of a pack.

In the drawing, elementary data such as video data and audio data is sequentially stored in a data storage area of a packet called a payload. The payload is

accompanied by a packet header. The packet header and the payload constitute one packet. The packet header stores information showing which type of stream, i.e., video or audio, the data in the payload belongs to. If there are a plurality of video streams or a plurality of audio streams, the packet header stores an ID (stream_id) for identifying which of these streams the data belongs to. The packet header also stores time stamps DTS and PTS which are information about decode time and presentation time of the data in the payload. Note here that the PTS and the DTS are not necessarily included in every packet header. MPEG defines a rule for storing the PTS and the DTS. Details on this rule are described in the MPEG system (ISO/IEC 13818-1) standard, and so its explanation has been omitted here.

The packet is accompanied by a header (pack header). The pack header and the packet constitute one pack. The pack header stores a time stamp SCR (System Clock Reference) showing when the pack is processed by a demultiplexer and is input in a decode buffer of the corresponding elementary stream.

(Interleaved Recording of a VOB)

The following describes interleaved recording of a VOB file, with reference to FIGS. 11 and 12.

The upper part of FIG. 11 shows part of the player construction described above. As illustrated, data on the BD disc is read via the optical pickup. If the data is a VOB, i.e., an MPEG stream, it is input in the track buffer. If the data is a PNG, i.e., image data, it is input in the image memory.

The track buffer is a FIFO buffer. Accordingly, the VOB data input in the track buffer is sent to the demultiplexer in the input order. Here, each pack is extracted from the track buffer according to the aforementioned SCR, and data in the pack is sent to the video processor or the audio processor via the demultiplexer. In the case of image data, on the other hand, the presentation controller indicates which image is to be drawn. Also, if the image data is for a subtitle, the image data is deleted from the image memory once it has been drawn. However, if the image data is for a menu, the image data is retained in the image memory even after it has been drawn. This is because drawing of a menu partially depends on user operations and so there is a possibility that the same image may be drawn a plurality of times.

The lower part of FIG. 11 shows interleaved recording of VOB files and PNG files on the BD disc. On a ROM such as a CD-ROM or a DVD-ROM, AV data which forms

one continuous playback unit is usually recorded in one continuous sequence. If data is continuously recorded in this way, a drive simply reads data and send it to a player in sequence. However, if continuous data is divided into a plurality of data sequences which are recorded at different positions on a disc, a seek operation needs to be performed between individual data sequences. Data reading is paused during this seek operation. This may cause data supply to be suspended. In the case of HD-DVDs too, therefore, it is desirable to record a VOB file in one continuous recording area. However, HD-DVDs also have data, such as subtitle data, which is to be played back in sync with video data in a VOB, so that it is necessary to read the subtitle data from the BD disc along with the VOB file.

One method of reading the subtitle data is reading all image data (PNG files) for subtitles in bulk before the playback of the VOB starts. However, this method is not realistic, as it requires a large memory for temporarily storing such image data.

In view of this, this embodiment employs a method of dividing a VOB file into several blocks and interleaved-recording the VOB blocks with image data. The lower part of FIG. 11 illustrates such interleaved recording.

By appropriately interleaving the VOB file and the image data, the image data can be stored in the image memory when necessary, with there being no need to use a large temporary memory mentioned above. In this case, however, the reading of the VOB data has to be suspended while the image data is being read.

FIG. 12 is a diagram for explaining a VOB data continuous supply model using a track buffer that solves this problem.

As described above, VOB data read from the BD disc is first stored in the track buffer. If a difference is provided between the data input rate and data output rate of the track buffer, the amount of data in the track buffer increases so long as data is being read from the BD disc.

Let Va be the input rate of the track buffer, and Vb the output rate of the track buffer. Also, suppose a continuous recording area for a VOB is from logical address a1 to logical address a2, as shown in the upper part of FIG. 12. Image data is recorded from a2 to a3, during which VOB data cannot be read.

The lower part of FIG. 12 shows the storage of the track buffer. The horizontal axis represents time, whereas the vertical axis represents the amount of data accumulated in the track buffer. Time t1 is the time

at which the reading of data at a1 that is the start point of the VOB continuous recording area begins. After this time, data is accumulated in the track buffer at the rate of Va-Vb. This rate is the difference between the input and output rates of the track buffer. Time t2 is the time at which data at a2 that is the end point of the VOB continuous recording area is read. Which is to say, the amount of data in the track buffer increases at the rate of Va-Vb from t1 to t2. When B(t2) denotes the amount of data in the track buffer at t2, B(t2) can be expressed as:

$$B(t2)=(Va-Vb)\times(t2-t1) \quad \cdots\cdots \text{(Formula 1)}$$

After this, image data continues until address a3 on the BD disc. Therefore, the input to the track buffer is 0 during this time. Hence the amount of data in the track buffer decreases at the rate of -Vb. This lasts until address a3, that is, time t3.

Here, if the amount of data in the track buffer becomes 0 before t3, it means there is no VOB data to be supplied to the decoder, causing the playback of the VOB to stop. However, if data remains in the track buffer at t3, the playback of the VOB can be continued without a pause.

This condition can be expressed as:

$$B(t2) \geq -Vb \times (t3-t2) \quad \cdots\cdots \text{(Formula 2)}$$

Therefore, the image data need be positioned on the BD disc so as to satisfy Formula 2.

(Structure of Navigation Data)

The following describes the structure of navigation data (BD management information) of the HD-DVD, by referring to FIGS. 13 to 19.

FIG. 13 shows the inner structure of a VOB management information file (YYY.VOBI).

The VOB management information includes stream attribute information (Attribute) and a time map (TMAP) for a VOB. The stream attribute information is made up of a video attribute (Video) and audio attributes (Audio#0 to Audio#m). Since one VOB can contain a plurality of audio streams, the presence of data fields is shown by the number of audio streams (Number).

The following are fields included in the video attribute (Video), and possible values of these fields.

Compression Format (Coding): MPEG 1
MPEG 2

	MPEG 4
Resolution (Resolution):	1920×1080
	1280×720
	720×480
	720×565
Aspect Ratio (Aspect):	4:3
	16:9
Frame Rate (Framerate):	60
	59.94
	50
	30
	29.97
	25
	24

The following are fields included in the audio attribute (Audio), and possible values of these fields.

Compression Format (Coding):	AC3
	MPEG1
	MPEG2
	LPCM

Number of Channels (Ch): 1 to 8

Language Attribute (Language):

The time map (TMAP) is a table holding information for each VOBU. The time map stores the number of VOBUs of the VOB (Number), and VOBU information for each VOBU (VOBU#1 to VOBU#n). The VOBU information for each VOBU includes a playback time length of the VOBU (Duration) and a data size of the VOBU (Size).

FIG. 14 is a diagram for explaining the VOBU information in greater detail.

It is widely known that an MPEG stream has two aspects, that is, the time and the data size. For example, the audio compression standard AC3 compresses data at a fixed bit rate. Accordingly, the time/address relationship can be expressed by a linear equation. In the case of MPEG video data, on the other hand, individual frames have a fixed display time period. As one example, each frame has a display time period of 1/29.97 second in NTSC. However, the data size of each frame after compression differs greatly, depending on the characteristics of the picture or the picture type used for compression, namely, I-picture, P-picture, or B-

picture. For MPEG video, therefore, the time/address relationship cannot be expressed by a general equation.

Accordingly, for a VOB that is an MPEG system stream in which MPEG video data has been multiplexed, the relationship between time and data cannot be expressed by a general equation. Instead, the time map (TMAP) serves to show the time/address relationship within the VOB. As shown in FIG. 14, the time map (TMAP) is a table that has two entries, i.e., the number of frames and the number of packs, for each VOBU.

A method of using such a time map (TMAP) is explained with reference to FIG. 15.

When time information is given as shown in FIG. 15, first a VOBU to which the time shown by the time information belongs is detected. This can be done by adding the number of frames of each VOBU one by one until a VOBU at which the total number of frames exceeds or matches the time (converted into the number of frames) is found. Next, the size of each VOBU preceding the detected VOBU in the time map is added together. The obtained sum is the start address of a pack which is to be read for playing a frame that contains the given time.

The following describes the inner structure of playlist information (XXX.PL), with reference to FIG. 16.

The playlist information is made up of a cell list (CellList) and an event list (EventList).

The cell list (CellList) is a playback cell sequence in a playlist. Cells are played back in the order shown by this cell list. The cell list (CellList) contains the number of cells (Number) and cell information for each cell (Cell#1 to Cell#n).

The cell information (Cell#) includes a VOB file name (VOBName), a valid section start time (In) and valid section end time (Out) within the VOB, and a subtitle table (SubtitleTable). The valid section start time (In) and the valid section end time (Out) are each shown by a frame number within the VOB. An address of VOB data to be played back can be obtained with reference to the aforementioned time map (TMAP).

The subtitle table (SubtitleTable) is a table holding subtitle information which is to be played in sync with the VOB. Like audio, subtitles can be provided in a plurality of languages. Accordingly, the subtitle table (SubtitleTable) includes the number of languages (Number) and a table for each language (Language#1 to Language#k) that follows.

Each language table (Language#) includes language information (Language), the number of sets of subtitle information for individually-displayed subtitles

(Number), and subtitle information for each individual subtitle (Speech#1 to Speech#j). The subtitle information (Speech#) is made up of a corresponding image data file name (Name), a subtitle display start time (In), a subtitle display end time (Out), and a subtitle display position (Position).

The event list (EventList) is a table defining events given in the playlist. The event list is made up of the number of events (Number) and the individual events (Event#1 to Event#m) that follow. Each event (Event#) is made up of an event type (Type), an event ID (ID), an event generation time (Time), and a valid time period (Duration).

FIG. 17 shows an event handler table (XXX.PROG) which holds event handlers (for time events and menu selection user events) for a playlist.

The event handler table has the number of event handlers / programs which have been defined (Number) and the individual event handlers / programs (Program#1 to Program#n). Each event handler / program (Program#) includes a definition of the start of the event handler (<event_handler> tag) and an event handler ID (ID) corresponding to an event ID. Following this, the program is written in parentheses "{" and "}" after "Function".

The following describes the internal structure of information relating to the overall BD disc (BD.INFO), with reference to FIG. 18.

The BD disc overall information is made up of a title list (TitleList) and an event table for global events (EventTable).

The title list (TitleList) includes the number of titles in the disc (Number) and title information for each title (Title#1 to Title#n) that follows. The title information (Title#) includes a table of playlists included in the title (PLTable) and a list of chapters in the title (ChapterList). The playlist table (PLTable) includes the number of playlists in the title (Number) and the names of the playlists, i.e., the file names of the playlists (Name).

The chapter list (ChapterList) includes the number of chapters included in the title (Number) and chapter information for each chapter (Chapter#1 to Chapter#n). The chapter information (Chapter#) includes a table of cells included in the chapter (CellTable). The cell table (CellTable) is made up of the number of cells (Number) and entry information for each cell (CellEntry#1 to CellEntry#k). The entry information (CellEntry#) has the name of a playlist that includes the cell, and the number of the cell in that playlist.

The Event list (EventList) includes the number of global events (Number) and information for each global event. It should be noted here that a global event which is defined first is a first event (FirstEvent). When the BD disc is inserted in the player, the first event is called first. The global event information is made up of an event type (Type) and an event ID (ID).

FIG. 19 shows a table of programs which are global event handlers (BD.PROG).

This table has the same structure as the event handler table shown in FIG. 17.

(Event Generation Mechanism)

The following describes a mechanism for generating an event, with reference to FIGS. 20 to 22.

FIG. 20 shows an example of a time event.

As explained earlier, a time event is defined by the event list (EventList) in the playlist information (XXX.PL). When the event type (Type) is TimeEvent, a time event having an ID "Ex1" is sent from the scenario processor to the program processor when event generation time (t1) is reached. The program processor searches for an event handler having the event ID "Ex1", and executes the event handler. In the present example, drawing of two button images is carried out.

FIG. 21 shows an example of a menu selection user event.

As described earlier, a menu selection user event is defined by the event list (EventList) in the playlist information (XXX.PL), like a time event. When the event type (Type) is UserEvent, a user event is set in the ready state when event generation time (t1) is reached. At this time, the event itself has not yet been generated. This event is in the ready state during a period shown by valid period information (Duration).

As shown in FIG. 21, when the user presses the Up/Down/Left/Right keys and the Return key on the remote controller, first a UOP event is generated by the UOP manager and sent to the program processor. The program processor passes the UOP event to the scenario processor. The scenario processor searches for a valid user event upon receiving the UOP event. If there is a valid user event, the scenario processor generates the user event, and sends it to the program processor. The program processor searches for an event handler that has an event ID "Ev1", and executes the event handler. In the present example, the playback of the playlist#2 is launched.

The generated user event does not contain information about which remote control key was pressed by the user. This information is given to the program

processor via the UOP event, and stored in a register SPRM (8) in a virtual player. The event handler program can refer to the value of this register and execute a branch.

FIG. 22 shows an example of a global event.

As described above, a global event is defined by the event list (EventList) in the BD disc overall information (BD.INFO). A global event, i.e., an event whose event type (Type) is GlobalEvent, is generated only when the user operates a key on the remote controller.

When the user presses the Menu key, first a UOP event is generated by the UOP manager and sent to the program processor. The program processor passes the UOP event to the scenario processor. The scenario processor generates a corresponding global event, and sends it to the program processor. The program processor searches for an event handler that has an event ID "menu", and executes the event handler. In the present example, the playback of the playlist#3 is launched.

Though this embodiment refers to a single Menu key, a plurality of Menu keys may be provided like DVDs. In such a case, an ID needs to be defined for each Menu key.

(Virtual Player Machine)

The following describes the functional

construction of the program processor, by referring to FIG. 23.

The program processor is a processing module that contains a virtual player machine. The virtual player machine is a functional model defined for HD-DVDs, and does not depend on HD-DVD players. Which is to say, the virtual player machine offers the same feature in any HD-DVD player.

The virtual player machine has two main features, namely, programming functions and player variables (registers). The programming functions define the following features as HD-DVD unique functions, based on JavaScript.

Link function: stop the current playback and start playback at a designated playlist, cell, and time.

Link (PL#, Cell#, time)

PL#: playlist name

Cell#: cell number

time: playback start time in the cell

PNG draw function: draw designated PNG data on the image plane.

Draw (File, X, Y)

File: PNG file name

X: X coordinate position

Y: Y coordinate position

Image plane clear function: clear a designated area of the image plane.

Clear (X, Y, W, H)

X: X coordinate position

Y: Y coordinate position

W: width in the X direction

H: width in the Y direction

The player variables include system parameters (SPRMs) which show the status of the player and general parameters (GPRMs) which can be used for general purposes.

FIG. 24 shows a list of system parameters (SPRMs).

SPRM(0): Language code

SPRM(1): Audio stream number

SPRM(2): Subtitle stream number

SPRM(3): Angle number

SPRM(4): Title number

SPRM(5): Chapter number

SPRM(6): Program number

SPRM(7): Cell number

SPRM(8): Selected key information

SPRM(9): Navigation timer

SPRM(10): Playback time information

SPRM(11): Mixing mode for karaoke

SPRM(12): Country information for parental management

SPRM(13): Parental level

SPRM(14): Player set value (video)

SPRM(15): Player set value (audio)

SPRM(16): Language code for audio stream

SPRM(17): Language code for audio stream (extension)

SPRM(18): Language code for subtitle stream

SPRM(19): Language code for subtitle stream (extension)

SPRM(20): Player region code

SPRM(21): reserved

SPRM(22): reserved

SPRM(23): Playback status

SPRM(24): reserved

SPRM(25): reserved

SPRM(26): reserved

SPRM(27): reserved

SPRM(28): reserved

SPRM(29): reserved

SPRM(30): reserved

SPRM(31): reserved

This embodiment describes the case where the programming functions of the virtual player are based on JavaScript, but other programming functions based on B-Shell or Perl Script used in UNIX OS and the like may also be used. In other words, the invention should not be limited to JavaScript.

(Program Examples)

FIGS. 25 and 26 show examples of event handler programs.

FIG. 25 shows an example of a menu which has two buttons.

A program on the left of FIG. 25 is executed using a time event, at the start of a cell (PlayList#1, Cell#1). Here, the general parameter GPRM(0) is initially set at 1. GPRM(0) is used to identify a selected button in this program. In the initial state, GPRM(0) indicates that button 1 on the left is being selected.

Next, PNG drawing is performed for each of button 1 and button 2, using the Draw function. For button 1, a PNG image "1black.png" is drawn with the coordinates (10, 200) as the start point (left end). For button 2,

a PNG image "2white.png" is drawn with the coordinates (330, 200) as the start point (left end).

Also, a program on the right of FIG. 25 is executed using a time event, at the end of the cell. Here, the Link function is used to instruct to play back from the start of the cell again.

FIG. 26 shows an example of an event handler for a menu selection user event.

The event handler includes a program corresponding to each operation using the Left key, Right key, and Return key on the remote controller. When the user presses any of the remote control keys, a user event is generated and the event handler of FIG. 26 is activated, as explained in FIG. 21. The event handler executes a branch, using the value of GRPM(0) identifying the selected button and the value of SPRM(8) identifying the pressed remote control key.

Condition 1): The Right key is pressed while button 1 is selected.

In this case, GPRM(0) is changed to 2, to put right button 2 in the selected state.

Also, the image of each of buttons 1 and 2 is redrawn.

Condition 2): The Return (OK) key is pressed while button 1 is selected.
In this case, the playback of the playlist#2 is launched.

Condition 3): The Return (OK) key is pressed while button 2 is selected.
In this case, the playback of the playlist#3 is launched.

Thus, programs are executed.

(Player Processing Flows)

The following describes processing flows of the player, by referring to FIGS. 27 to 30.

FIG. 27 is a flowchart showing basic processing up to AV playback.

When the BD disc is inserted (S101), the HD-DVD player reads and analyzes the BD.INFO file (S102), and reads the BD.PROG file (S103). Here, BD.INFO and BD.PROG are stored in the management information storage memory, and then analyzed by the scenario processor.

Following this, the scenario processor generates the first event, according to the first event (FirstEvent) information in the BD.INFO file (S104).

The program processor receives the first event, and executes an event handler corresponding to the first event (S105).

The event handler corresponding to the first event is expected to contain information of a playlist to be played first. If a playlist is not designated in the event handler, the player has nothing to play back, and has to wait for receiving a user event. Therefore, when a playlist is not designated in the event handler, the player waits for a user event (S201). Upon receiving the user's remote control operation, the UOP manager sends a UOP event to the program manager (S202).

The program manager judges whether the UOP event is made by the Menu key (S203). If so, the program manger sends the UOP event to the scenario processor, and the scenario processor generates a user event (S204). The program processor executes an event handler corresponding the user event (S205).

FIG. 28 is a flowchart showing processing from the start of PL playback to the start of VOB playback.

As described above, playback of a playlist is launched by the first event handler or a global event handler (S301). The scenario processor reads and analyzes playlist information XXX.PL which is necessary to play back the playlist (S302). The scenario

processor also reads program information XXX.PROG corresponding to the playlist (S303). After this, the scenario processor starts playback of a cell, based on cell information shown in the playlist (S304). To play back the cell, the scenario processor issues an instruction to the presentation controller, which accordingly starts playing AV data (S305).

When the playback of the AV data starts (S401), the presentation controller reads and analyzes a VOB information file (YYY.VOBI) of a VOB corresponding to the cell (S402). The presentation controller specifies a VOBU to be played and its address, using a time map. The presentation controller then indicates the read address to the drive controller. The drive controller reads VOB data from the indicated address (S403), and sends it to the decoder. Thus, the playback of the VOB data begins (S404).

The VOB playback continues until the end of the playback section of the VOB specified by the cell (S405). After the end of the playback section, the processing proceeds to step S304 to play back the next cell. If the next cell does not exist, the playback operation ends (S406).

FIG. 29 is a flowchart showing event processing after the start of AV playback.

The HD-DVD player is an event-driven player model. Once playback of a playlist begins, event processes for time events, user events, and subtitle displays are activated, so that the HD-DVD player executes these event processes in parallel.

Steps S501 to S505 correspond to the time event processing.

After the playback of the playlist starts (S501), the scenario processor judges whether the playlist playback has completed (S502). If not, the scenario processor judges whether time event generation time is reached (S503). When the time event generation time is reached, the scenario processor generates a time event (S504). The program processor receives the time event, and executes an event handler (S505).

If the time event generation time is not reached in step S503 or after the event handler has been executed in step S505, the procedure returns to step S502 to repeat the above processing. If the playlist playback is completed in step S502, the time event processing is forced to terminate.

Steps S601 to S608 correspond to the user event processing.

After the playlist playback starts (S601), the completion of the playlist playback is judged (S602),

and then the reception of a UOP is judged (S603). Upon receiving a UOP, the UOP manager generates a UOP event (S604). The program processor receives the UOP event, and judges whether the UOP event is a menu call (S605). If it is, the program processor instructs the scenario processor to generate an event (S607). The program processor then executes an event handler (S608).

If the UOP event is not a menu call in step S605, it means the UOP event is an event caused by a cursor key or the Return key. In this case, the scenario processor judges whether the current time is within a user event valid duration (S606). If so, the scenario processor generates a user event (S607), and the program processor executes an event handler (S608).

If a UOP is not received in step S603, if the current time is not within the user event valid duration in step S606, or after the event handler has been executed in step S608, the procedure returns to step S602 to repeat the above processing. Also, if the playlist playback is completed in step S602, the user event processing is forced to terminate.

FIG. 30 corresponds to the subtitle processing.

After the playlist playback starts (S701), the completion of the playlist playback is judged (S702), and then the arrival of the start time for drawing a

subtitle is judged (S703). When the subtitle draw start time is reached, the scenario processor instructs the presentation controller to draw the subtitle. The presentation controller instructs the image processor to draw the subtitle (S704). If the subtitle draw start time is not reached in step S703, it is judged whether subtitle display end time is reached (S705). If the subtitle display end time is reached, the presentation controller instructs the image processor to erase the subtitle, and the image processor erases the subtitle from the image plane (S706).

After the subtitle drawing step S704 has ended, after the subtitle erasing step S706 has ended, or if the subtitle display end time is not reached in step S705, the procedure returns to step S702 to repeat the above processing. If the playlist playback is completed in step S702, the subtitle processing is forced to terminate.

(Second Embodiment)

The following describes the second embodiment of the present invention.

The second embodiment relates to a BD player which has a plurality of playback/execution systems. Since the second embodiment is fundamentally based on the first

embodiment, the following description focuses on the developments or differences from the first embodiment.

FIG. 31 compares the world of BD applications with the world of DVD applications.

The DVD world is a closed world with a DVD disc and a DVD player. The DVD disc stores AV data and navigation information necessary for playback control. The DVD player has a playback control program, and processes static data of the DVD disc using the playback control program.

On the other hand, the BD (HD-DVD) world is a world linked with the Internet so that new content or a new execution program can be downloaded and played/executed by a BD player (HD-DVD player), as can be seen from FIG. 31. The BD player has several playback modes. The user can enjoy video content such as movies, through a variety of viewing modes such as a DVD-compatible playback mode, a browser mode, and a Java mode.

Java (TM) is middleware developed by Sun Microsystems. In recent years, Java has been widely used for consumer devices such as mobile phones in Japan and digital broadcasting DVB-MHP in Europe. Java is an object-oriented programming language similar to C++ and the like. However, while C++ depends on the execution environment, i.e., the type of OS, Java defines a virtual

machine (Java Virtual Machine) which exists for most OSs including Windows and Linux. Hence Java is receiving attention as an OS-independent execution system. Even when the execution environment differs for each manufacturer, as likely to be the case for consumer devices, an application program can be executed regardless of the execution environment. For this reason, Java is employed in mobiles phones and STBs.

BDs (HD-DVDs) can greatly benefit users by employing Java, too. For example, the same game can be played between BD players of different manufacturers.

FIG. 32 shows the structure of a BD in the second embodiment.

According to this structure, other media such as an HDD, a memory card, and a network are provided in the lowest layer in addition to a BD medium. This indicates that logical data of the BD can be recorded not only on the physical BD medium but also on various other media.

When compared with FIG. 4 of the first embodiment, the remarkable characteristics of the BD in this embodiment are that the aforementioned different playback modes are provided in the highest layer. This enables BD content to be played while switching between various modes such as the DVD-compatible playback mode (corresponding to the first embodiment), the browser

playback mode, and the Java playback mode.

A BD basic playback feature enables the playback to be switched between these different modes. This layer contains a playback control feature which is a basic feature necessary for playing the BD, and a player status.

FIG. 33 is a conceptual diagram of the BD basic playback feature.

As shown in the drawing, actual playback of a playlist is conducted by the BD basic playback feature, regardless of whether the playback mode is DVD-compatible or Java. Which is to say, the BD basic playback feature has an I/F for each playback mode such as the DVD-compatible playback mode. Each playback mode uses a corresponding I/F to indicate playback of a playlist or to acquire a language attribute.

Each playback mode, such as the DVD-compatible playback mode explained in the first embodiment or the Java playback mode, offers programming functions for describing dynamic scenarios. Some programming functions such as for executing playback of a playlist are realized in the layer of the BD basic playback feature.

FIG. 34 shows the directory/file structure on the BD disc. A special directory BDVIDEO is provided

immediately below a ROOT directory, and files necessary for playing the BD are provided immediately below the BDVIDEO directory, as in the first embodiment.

The difference with the first embodiment lies in that a JCLASSES directory is provided immediately below the BDVIDEO directory as a subdirectory. This directory contains Java application classes used for playing the BD.

In FIG. 34, BD.CLASS is a Java class file. This class file is executed by a Java player program provided in the BD player. Other class files are called from BD.CLASS.

FIG. 35 shows the construction of a player in the second embodiment. The following focuses on the difference from the first embodiment.

The difference from the BD player construction of the first embodiment shown in FIG. 7 lies in the construction of the program processor (302). In the BD player of the second embodiment, the program processor (302) includes a DVD-compatible module (3021), a browser module (3022), a Java module (3023), a playback controller (3024), and a BD basic feature processor / BD-FF processor (3025).

The DVD-compatible module (3021), the browser module (3022), and the Java module (3023) are switchable

modules. These modules are managed and controlled by the playback controller (3024). When the Java class file BD.CLASS is provided on the BD disc, the playback controller (3024) selects the Java module (3023). The Java module (3023) reads the Java class file BD.CLASS which is read from the BD disc and stored in the program storage memory (301), and executes it.

The BD-FF processor (3025) is a module for realizing the aforementioned BD basic playback feature. To realize playback control, actually each of the DVD-compatible module (3021), the browser module (3022), and the Java module (3023) calls the feature of the BD-FF processor (3025).

FIG. 36 shows the structure of a virtual player machine in the second embodiment.

As noted earlier, the virtual player machine of the second embodiment has a hierarchy. The BD-FF processor which provides the BD basic playback feature is placed in the lowest layer, whilst the DVD-compatible playback feature, the browser playback feature, and the Java playback feature are placed in the highest layer.

The BD-FF processor has the following features necessary for control of BD playback, and parameters showing the player status.

Link feature: capable of designating the following parameters.

PL#: playlist name

Cell#: cell number

time: playback start time in the cell

PNG draw feature: capable of designating the following parameters.

File: PNG file name

X: X coordinate position

Y: Y coordinate position

Image plane clear feature: capable of designating the following parameters.

Clear(X, Y, W, H)

X: X coordinate position

Y: Y coordinate position

W: width in the X direction

H: width in the Y direction

32 system parameters (SPRMs) showing the status of the player.

SPRM(0) to SPRM(31): individual values are the same as those in the first embodiment.

16 general parameters (GPRMs) which can be used for general purposes.

GPRM(0) to GRPM()

In the case of the DVD-compatible playback feature, these features are expressed as the functions and variables of JavaScript, and can be used by event handler programs as explained in the first embodiment.

In the case of the Java playback feature, on the other hand, they are expressed as BD special classes as shown in FIG. 36, and can be called from application programs. The BD special classes are the following.

1) Status class: BdStatus

The BdStatus class is automatically generated by a BD player object. The status of the player can be acquired or set using member functions of the BdStatus class.

The BdStatus class has the following member functions.

Language code acquisition / setting function

getLanguage()

setLanguage()

Audio stream number acquisition / setting function

getAudioStreamNumber()

setAudioStreamNumber()

Subtitle stream number acquisition / setting function

getSubtitleStreamNumber()

setSubtitleStreamNumber()

Angle number acquisition / setting function

getAngleNumber()

setAngleNumber()

Title number acquisition function

getTitleNumber()

Chapter number acquisition function

getChapterNumber()

Program number acquisition function

getProgramNumber()

Cell number acquisition function

getCellNumber()

Selected key information acquisition function

getExecutedKeyInfo()

Navigation timer acquisition function

getNavigationTimerValue()

Playback time information acquisition function

getCurrentElapseTime()

Karaoke mixing mode acquisition function

getKARAOKEMixingMode()

Parental country information acquisition function

getParentalCountryInfo()

Parental level acquisition function

getParentalLevel()

Player set value (video) acquisition function

getPlayerConfigurationForVideo()

Player set value (audio) acquisition function

getPlayerConfigurationForAudio()

Audio stream language code acquisition function

getLanguageForAudio()

Subtitle stream language code acquisition function

getLanguageForSubtitle()

Player region code acquisition function

getRegionCode()

Playback status acquisition function

getPlaybackStatus()

2) Playback class: BdPlayback

The BdPlayback class is automatically generated by the BD player object. The playback control of the player can be carried out using member functions of the BdPlayback class.

The BdPlayback class has the following member

functions.

Playlist link function: start playback from a designated playlist, cell, and time.

linkPL (PL#, Cell#, time)

PL#: playlist name

Cell#: cell number

time: playback start time in the cell

PNG draw function: draw designated PNG data on the image plane.

drawImage (File, X, Y)

File: PNG file name

X: X coordinate position

Y: Y coordinate position

Image plane clear function: clear a designated area of the image plane

clearImage (X, Y, W, H)

X: X coordinate position

Y: Y coordinate position

W: width in the X direction

H: width in the Y direction

3) Event class: BdEvent

The BdEvent class is automatically generated by the BD player object. This class is generated at the timing of generating a user event and a time event explained in the first embodiment, when executing the Java playback feature. Which is to say, this corresponds to an object that is generated instead of a user event and a time event.

The BdEvent class has the following member function.

Setting function for event acquisition

setEventListener(this)

this: the object which calls the member function

FIG. 37 shows an example where the above Java class functions are actually used.

First, the player activates the BD player object. When BD.CLASS is present on the BD disc, the BD player object reads this class, and generates a BD object.

The BD object is an application program provided with the BD disc, and is not installed on the player side. The BD object has an event handler as its member function. After the BD object is generated, setEventListener which is the member function of the event object is called to

declare the availability of the event handler.

If a user event or time event explained in the first embodiment occurs, the event object activates the event handler of the BD object set by the setEventListener function.

The event handler achieves a dynamic scenario, by executing functions such as getLanguage() which is a member function of the player status object and link() which is a member function of the playback object.

Processing flows of the BD player are explained below, with reference to FIGS. 38 to 40.

Here, the explanation concerns only the DVD-compatible feature and the Java feature, among the aforementioned switchable features, namely, the DVD-compatible feature, the browser feature, and the Java feature. The browser feature is certainly one of the switchable features. However, the point of the present invention is to switch between a plurality of features, so that the three features are limited to the two for simplicity's sake.

FIG. 38 is a flowchart showing basic processing after a disc is inserted.

When the BD disc is inserted (S1001), the BD player judges whether the BD.CLASS file is present (S1002). If the BD.CLASS file is not present, the BD player proceeds

to step S102 in the first embodiment, to perform the same processing as in the first embodiment.

If the BD.CLASS file is present, a BD object which is an instance of BD.CLASS is generated by the Java module (S1003). The BD object calls the member function of the event object, i.e., setEventListener, and declares the event handler (S1004).

Following this, the BD player reads BD.INFO (S1005), and generates the first event, i.e., FirstEvent (S1006). As a result, the event handler of the BD object is executed by the event object (S1007).

FIG. 39 is a flowchart showing time event processing after playback of a playlist starts.

After playback of a playlist starts (S1101), the player judges whether the playback of the playlist has completed (S1102), and then judges whether time event execution time is reached (S1103). If so, the player generates a time event (S1104). As a result, the event handler of the BD object is executed by the event object (S1105).

If the time event time is not reached in step S1103 or after the event handler has been executed, the processing returns to step S1102 to judge the completion of the playlist playback.

FIG. 40 is a flowchart showing mode switching

processing of the BD player.

After the playlist playback starts (S1201), the player receives a request for switching the current mode (S1202). Step S1202 is repeated until a mode switching request is received from the user. Upon receiving a mode switching request, the player proceeds to a different step according to the current mode (S1203). If the current mode is the DVD-compatible mode, the playback controller switches the valid module from the DVD-compatible module to the Java module (S1204). If the current mode is the Java mode, the playback controller switches the valid module from the Java module to the DVD-compatible module (S1205).

Here, mode switching is carried out not by completely stopping an individual feature, but by sending an event sent from the scenario processor to one module so as to invalidate the dynamic scenario processing feature of the other module. In the construction of FIG. 35, this is achieved by the playback controller selecting one of the DVD-compatible module, the browser module, and the Java module and controlling the flow of an event.

(Other Modifications)

Although the present invention has been described

by way of the above embodiments, it should be obvious that the invention is not limited to the above. Example modifications are given below.

(1) Each of the above apparatuses is actually realized by a computer system including a microprocessor, a ROM, a RAM, and the like. In this case, a computer program is stored in the RAM, and the microprocessor operates in accordance with this computer program to achieve the above features.

(2) The present invention also applies to the above methods. These methods may be realized by a computer program that is executed by a computer. Such a computer program may be distributed as a digital signal.

The present invention may be realized by a computer-readable storage medium, such as a flexible disc, a hard disk, a CD-ROM, a MO, a DVD, a DVD-ROM, a DVD-RAM, a BD (Blu-ray Disc), or a semiconductor memory, on which the computer program or digital signal mentioned above is recorded. Conversely, the present invention may also be realized by a computer program or digital signal that is recorded on such a storage medium.

A computer program or digital signal that achieves the present invention may also be transmitted via a network, such as an electronic communications network, a wired or wireless communications network, or the

Internet.

The present invention can also be realized by a computer system that includes a microprocessor and a memory. In this case, a computer program can be stored in the memory, with the microprocessor operating in accordance with this computer program.

Also, the computer program or digital signal may be provided to an independent computer system by distributing a storage medium on which the computer program or digital signal is recorded, or by transmitting the computer program or digital signal via a network. The independent computer system may then execute the computer program or digital signal to function as the present invention.

(3) The limitations described in the embodiments and the modifications may be freely combined.

(Effects of the Invention)

According to the present invention, a player has a plurality of playback modules (DVD-compatible, browser, Java). This being so, information which can be shared by these different playback modules, such as player status information and playback control features, is provided as a BD basic playback feature independently of the playback modules. As a result, even when one

playback module is switched to another, the playback condition such as the player status need not be changed. This enables switching between different playback modes to be conducted easily.

(Numerical References)

201 .... BD disc
202 .... optical pickup
203 .... program storage memory
204 .... management information storage memory
205 .... AV storage memory
206 .... program processing unit
207 .... management information processing unit
208 .... presentation processing unit
209 .... image plane
210 .... video plane
211 .... superimposing unit

301 .... program storage memory
302 .... program processor
303 .... UOP manager
304 .... management information storage memory
305 .... scenario processor
306 .... presentation controller
307 .... clock

308 .... image memory

309 .... track buffer

310 .... demultiplexer

311 .... image processor

312 .... video processor

313 .... audio processor

314 .... image plane

315 .... video plane

316 .... superimposing unit

317 .... drive controller

3021 .... DVD-compatible module

3022 .... browser module

3023 .... Java module

3024 .... playback controller

3025 .... BD-FF processor

S101 .... disc inserting step

S102 .... BD.INFO reading step

S103 .... BD.PROG reading step

S104 .... first event generating step

S105 .... event handler executing step

S201 .... UOP receiving step

S202 .... UOP event generating step

S203 .... menu call judging step

S204 .... event generating step

S205 .... event handler executing step

S301 .... playlist playback starting step

S302 .... playlist information (XXX.PL) reading step

S303 .... playlist program (XXX.PROG) reading step

S304 .... cell playback starting step

S305 .... AV playback starting step

S401 .... AV playback starting step

S402 .... VOB information (YYY.VOBI) reading step

S403 .... VOB (YYY.VOB) reading step

S404 .... VOB playback starting step

S405 .... VOB playback ending step

S406 .... next cell presence judging step

S501 .... playlist playback starting step

S502 .... playlist playback end judging step

S503 .... time event time judging step

S504 .... event generating step

S505 .... event handler executing step

S601 .... playlist playback starting step

S602 .... playlist playback end judging step

S603 .... UOP reception judging step

S604 .... UOP event generating step

S605 .... menu call judging step

S606 .... user event valid duration judging step

S607 .... event generating step

S608 .... event handler executing step

S701 .... playlist playback starting step

S702 .... playlist playback end judging step

S703 .... subtitle draw start judging step

S704 .... subtitle drawing step

S705 .... subtitle display end judging step

S706 .... subtitle erasing step

S1001 .... disc inserting step

S1002 .... BD.CLASS detection judging step

S1003 .... BD object generating step

S1004 .... event handler declaring step

S1005 .... BD.INFO reading step

S1006 .... first event generating step

S1007 .... event handler executing step

S1101 .... playlist playback starting step

S1102 .... playlist end judging step

S1103 .... time event time judging step

S1004 .... time event generating step

S1005 .... event handler executing step

S1201 .... playlist playback starting step

S1202 .... mode switch judging step

S1203 .... valid mode judging step

S1204 .... Java mode switching step

S1205 .... DVD-compatible mode switching step

Although the present invention has been fully described by way of examples with reference to the accompanying drawings, it is to be noted that various changes and modifications will be apparent to those skilled in the art.

Therefore, unless such changes and modifications depart from the scope of the present invention, they should be construed as being included therein.

What is claimed is:

1. A playback apparatus for playing back data on an optical disc on which at least video data and audio data are recorded, comprising:

a plurality of application playback modules; and

a controller for selecting an application playback module from the plurality of application playback modules.

2. The playback apparatus of Claim 1, including:

a basic playback unit for performing a basic playback operation that is commonly used by the plurality of application playback modules.

3. The playback apparatus of one of Claims 1 and 2, wherein the basic playback unit includes:

a status storage unit for storing information showing a status of the playback apparatus; and

a playback control unit for controlling playback of the video data and the audio data.

4. A playback system comprising:

an optical disc on which at least video data and audio data are recorded; and

a playback apparatus for playing back data on the

optical disc, the playback apparatus including:

a plurality of application playback modules; and

a controller for automatically selecting, if an application module is stored in a predetermined directory on the optical disc, a predetermined application playback module from the plurality of application playback modules,

wherein the application module is activated by the selected application playback module.

5. A playback method for playing back data on an optical disc on which at least video data and audio data are recorded, comprising:

a plurality of application playback steps; and

a control step for selecting an application playback step from the plurality of application playback steps.

ABSTRACT OF DISCLOSURE

(Object)

In conventional DVD techniques, playback control of a DVD is limited to a single player feature. Since playback control information is embedded in a stream, it is difficult to freely switch the player feature to another or to modify the player feature.

(Solution)

A BD player includes a plurality of application playback modules, a controller for selecting one of the plurality of application playback modules, and a basic playback unit (BD-FF processor) which possesses, independently of each application playback module, a basic playback feature common to the plurality of application playback modules. In this way, an application playback module can be switched to another, while maintaining the playback status of the player.

(Selected Drawing)

FIG. 36

FIG.1

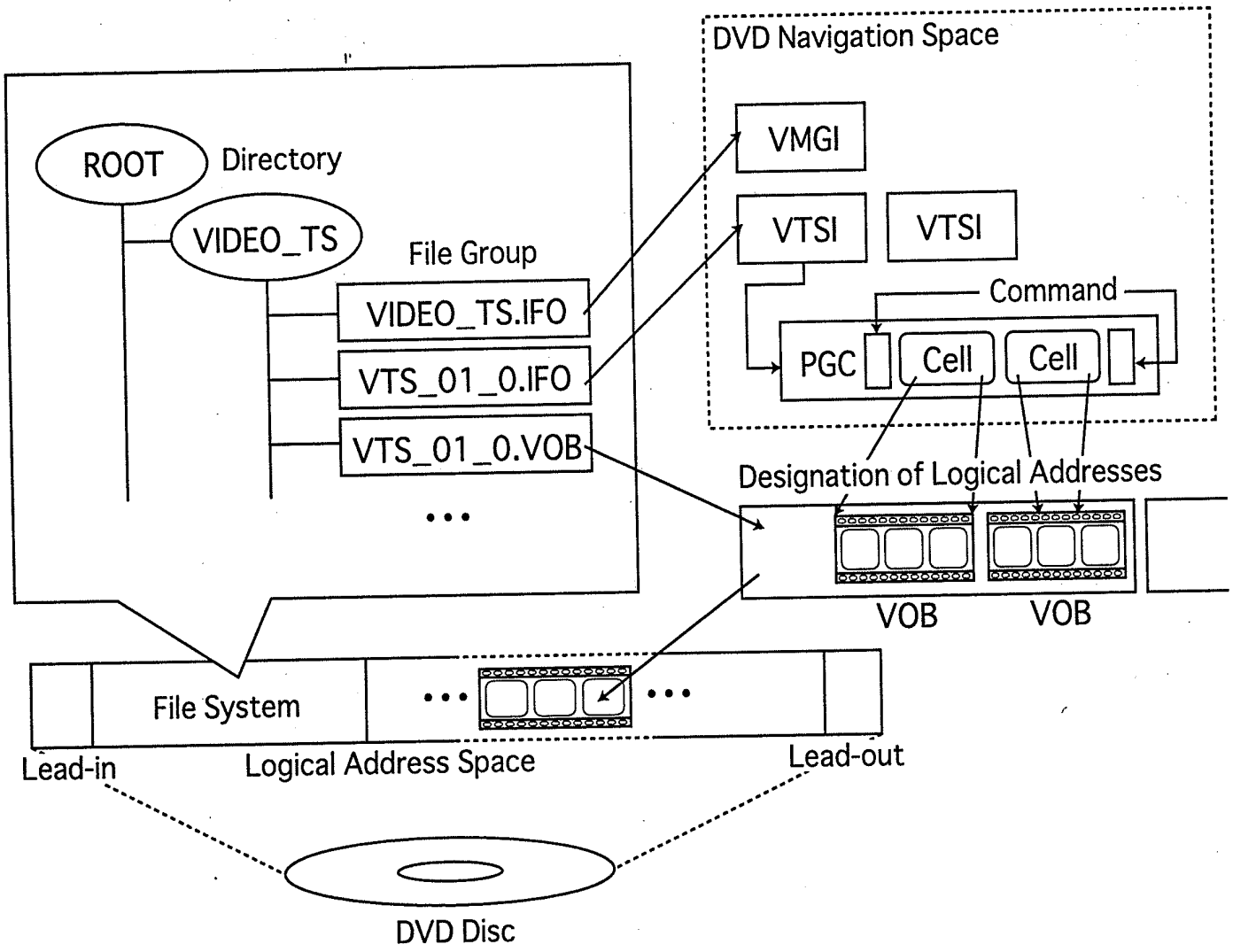

DVD Navigation Space
ROOT
Directory
VIDEO_TS
File Group
VIDEO_TS.IFO
VTS_01_0.IFO
VTS_01_0.VOB
VMGI
VTSI
VTSI
Command
PGC
Cell
Cell
Designation of Logical Addresses
VOB
VOB
File System
Lead-in
Logical Address Space
Lead-out
DVD Disc

FIG.2

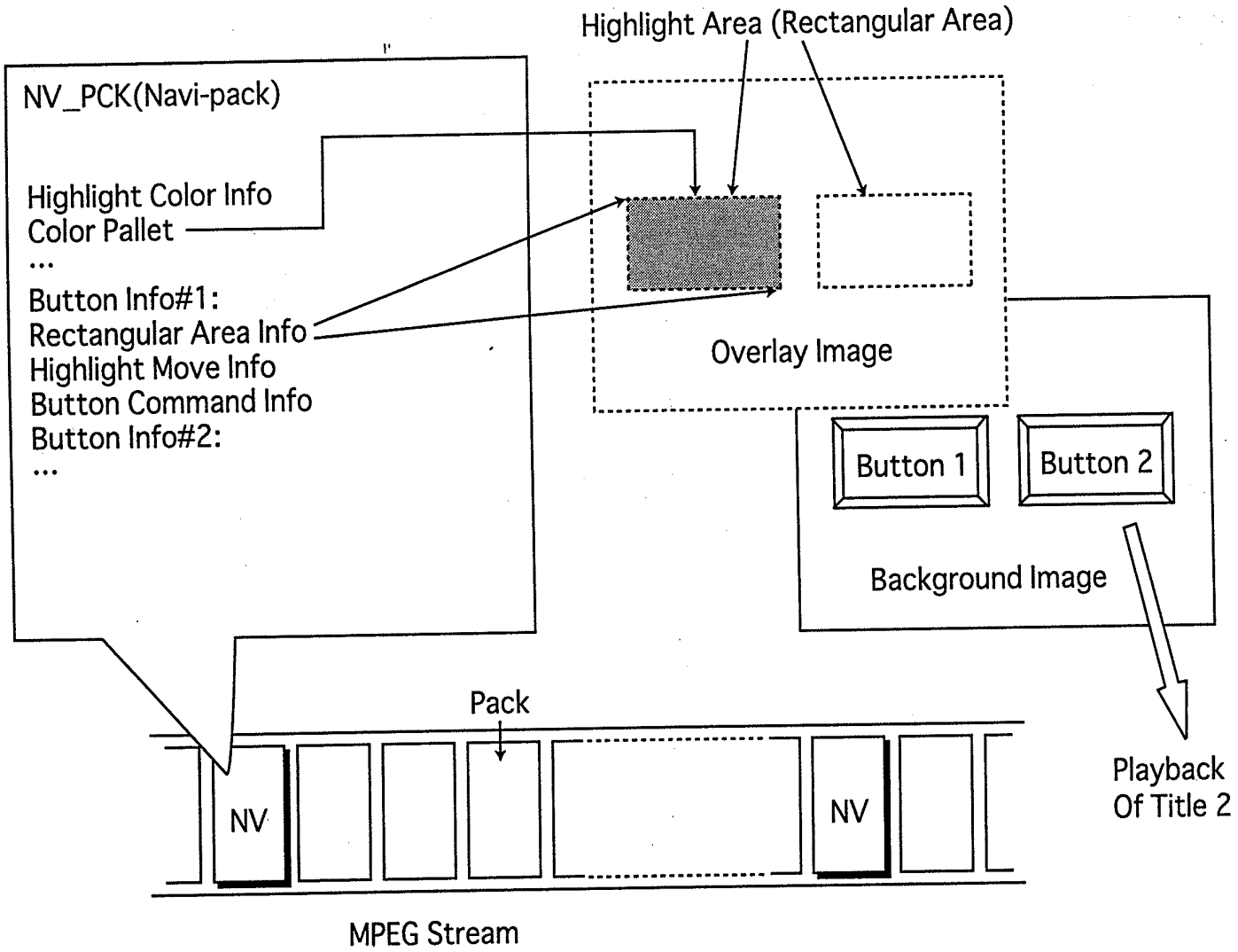

Highlight Area (Rectangular Area)
NV_PCK(Navi-pack)
Highlight Color Info
Color Pallet
...
Button Info#1:
Rectangular Area Info
Highlight Move Info
Button Command Info
Button Info#2:
...
Overlay Image
Button 1
Button 2
Background Image
Pack
NV
NV
Playback
Of Title 2
MPEG Stream

FIG.3

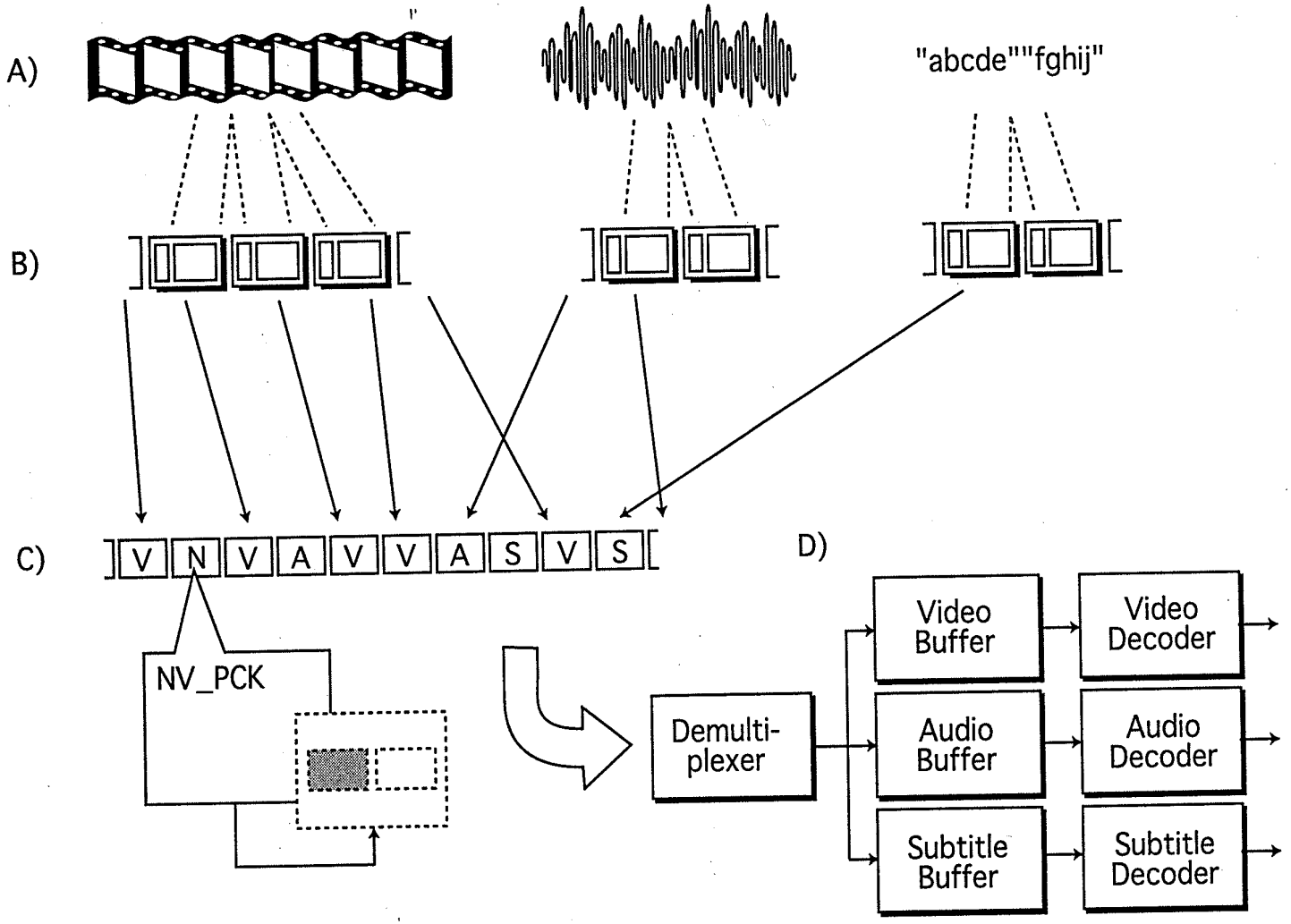

A)
"abcde""fghij"
B)
C)
V
N
V
A
V
V
A
S
V
S
D)
NV_PCK
Demulti-
plexer
Video
Buffer
Audio
Buffer
Subtitle
Buffer
Video
Decoder
Audio
Decoder
Subtitle
Decoder

FIG.4

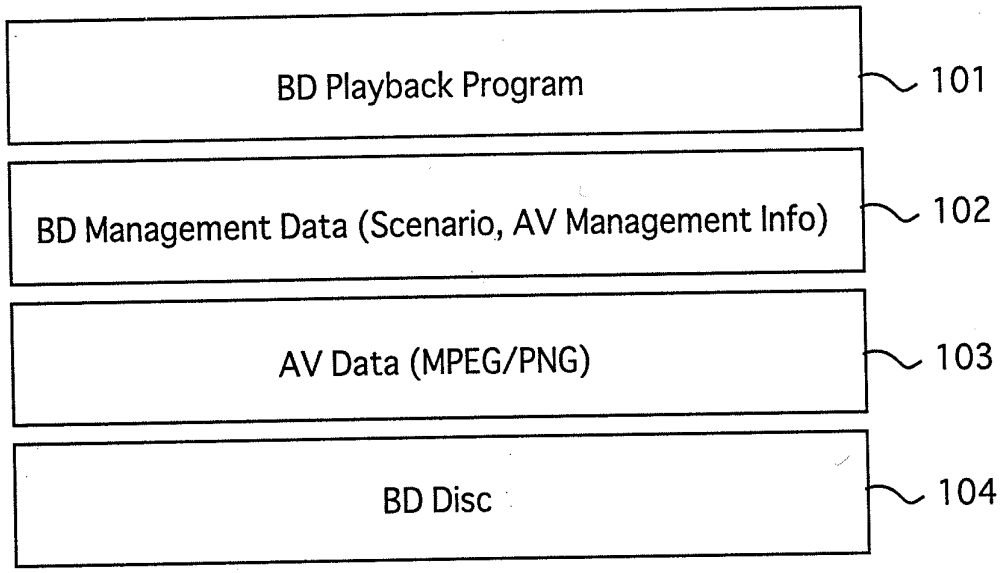

BD Playback Program
101
BD Management Data (Scenario, AV Management Info)
102
AV Data (MPEG/PNG)
103
BD Disc
104

FIG.5

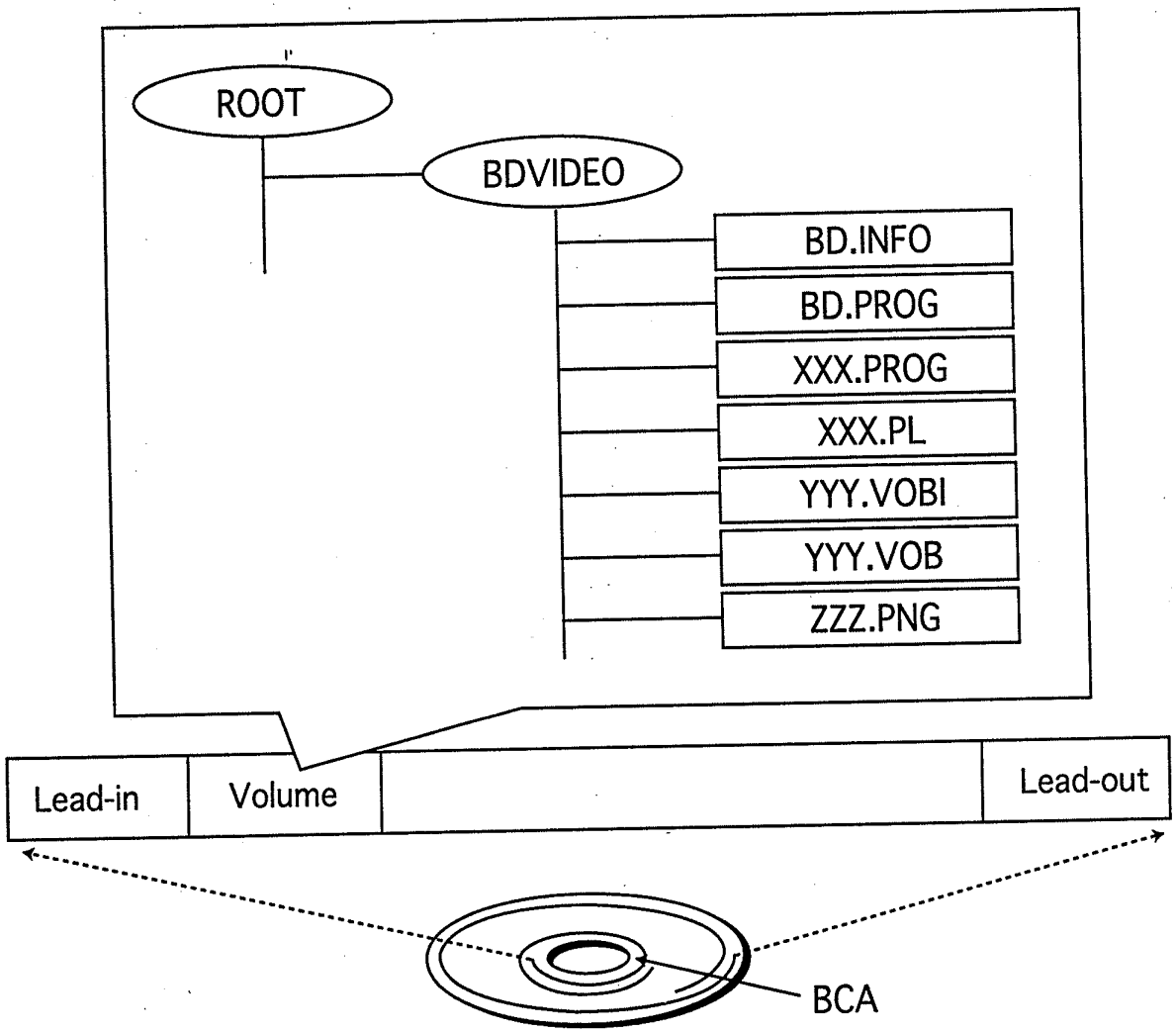

ROOT
BDVIDEO
BD.INFO
BD.PROG
XXX.PROG
XXX.PL
YYY.VOBI
YYY.VOB
ZZZ.PNG
Lead-in
Volume
Lead-out
BCA

FIG.6

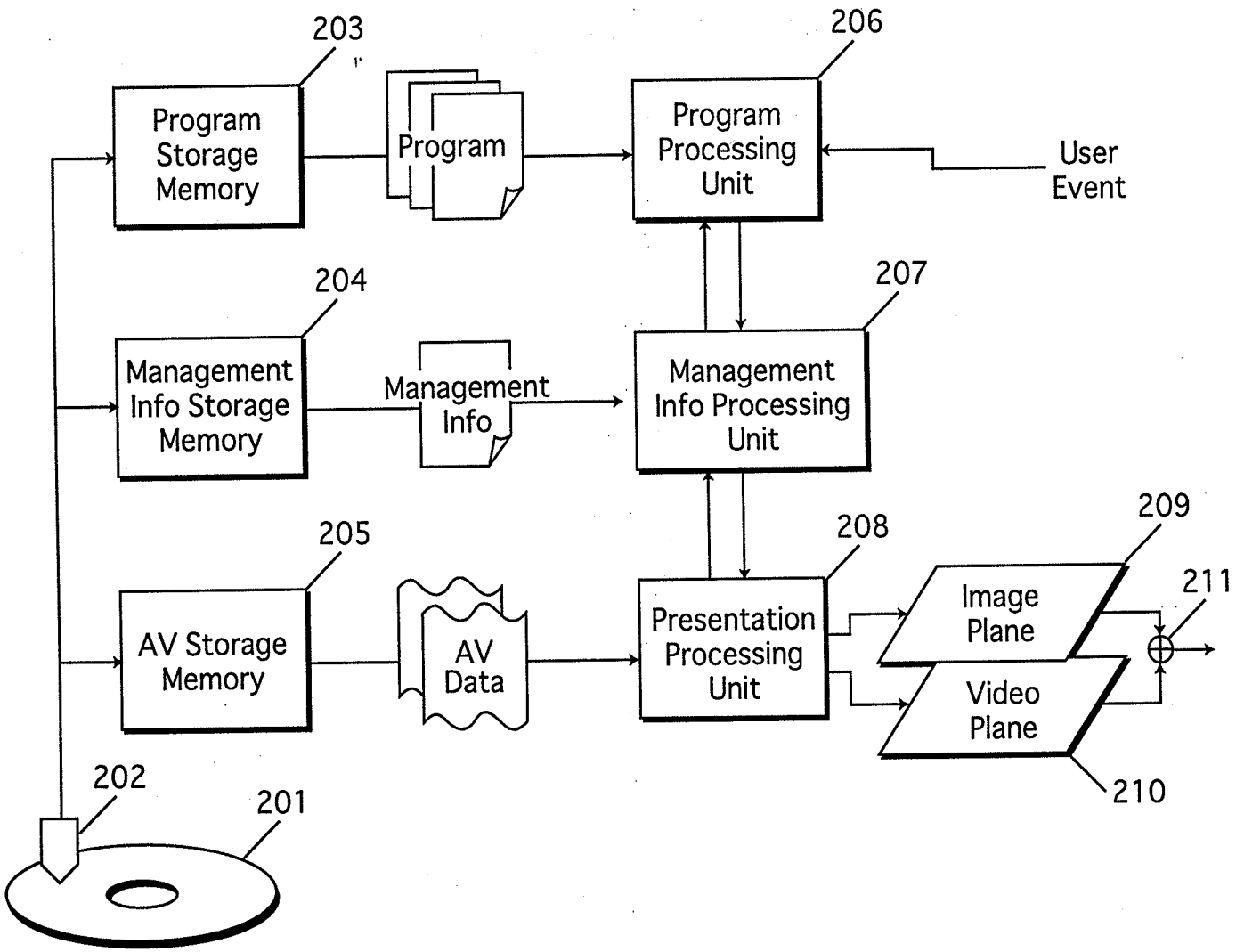

203
Program Storage Memory
Program
206
Program Processing Unit
User Event
204
Management Info Storage Memory
Management Info
207
Management Info Processing Unit
205
AV Storage Memory
AV Data
208
Presentation Processing Unit
209
Image Plane
211
Video Plane
210
202
201

FIG.7

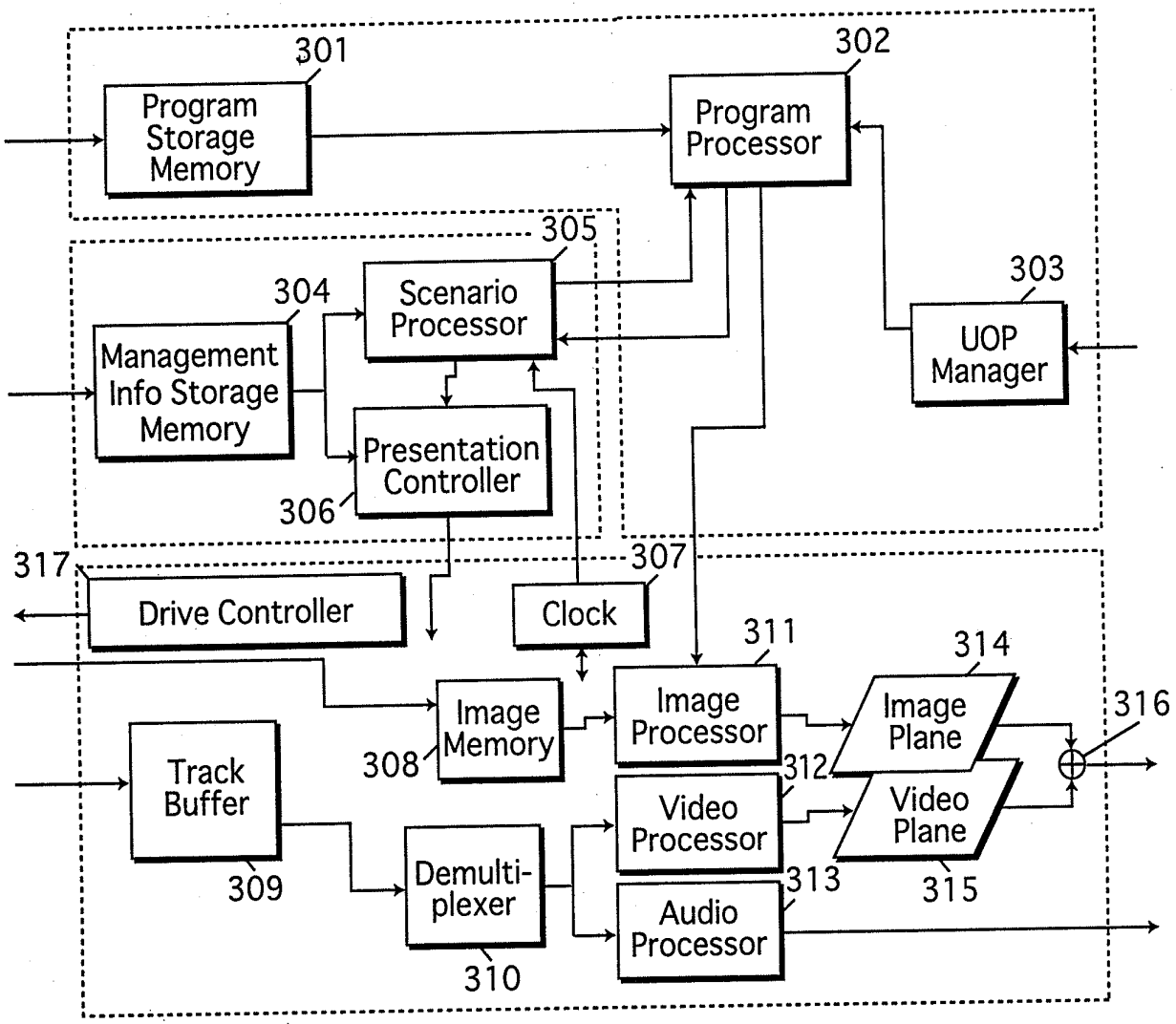

301
Program Storage Memory
302
Program Processor
305
304
Scenario Processor
303
UOP Manager
Management Info Storage Memory
Presentation Controller
306
317
307
Drive Controller
Clock
311
314
Image Processor
Image Plane
316
Image Memory
308
312
Track Buffer
Video Processor
Video Plane
309
Demulti-plexer
313
315
Audio Processor
310

FIG.8

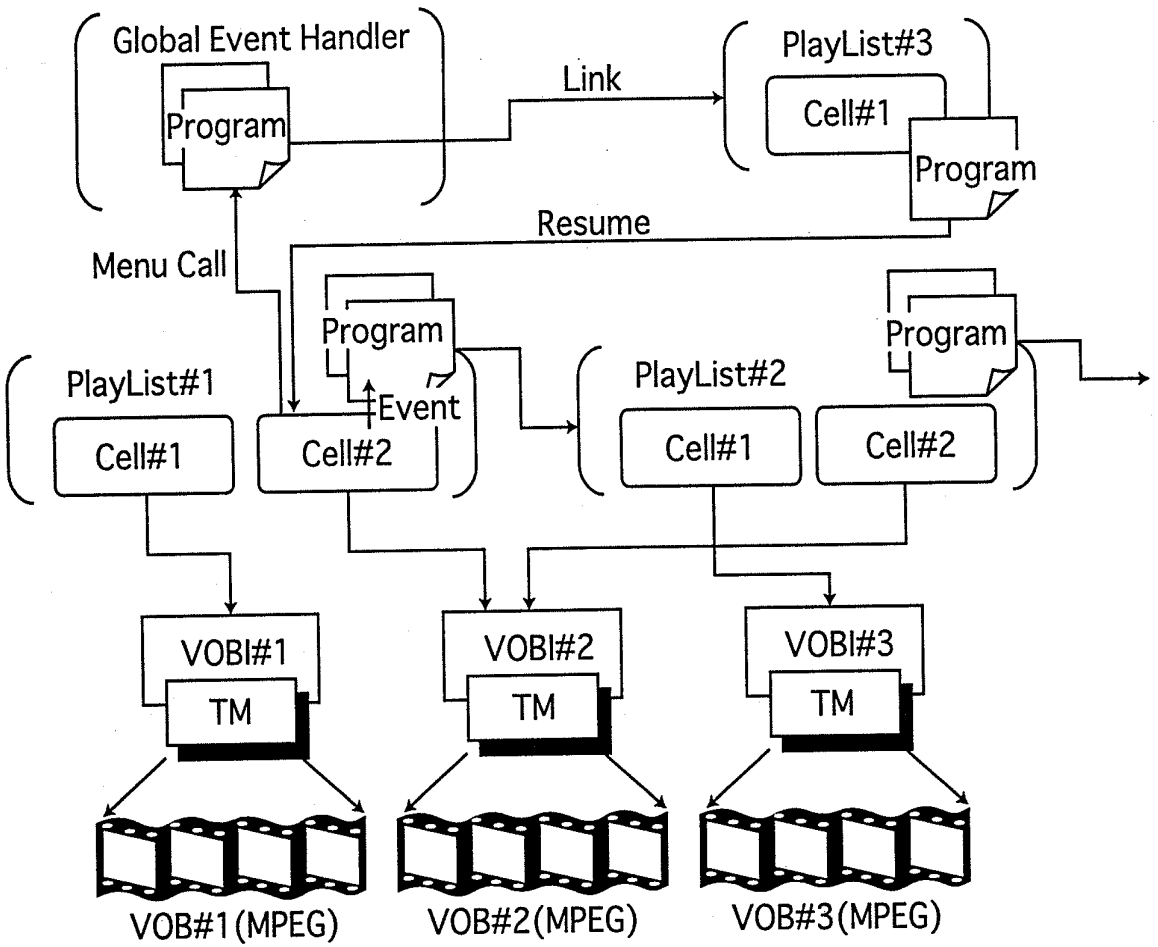

Global Event Handler
Program
Link
PlayList#3
Cell#1
Program
Resume
Menu Call
Program
Program
PlayList#1
Event
PlayList#2
Cell#1
Cell#2
Cell#1
Cell#2
VOBI#1
TM
VOBI#2
TM
VOBI#3
TM
VOB#1(MPEG)
VOB#2(MPEG)
VOB#3(MPEG)

FIG.9

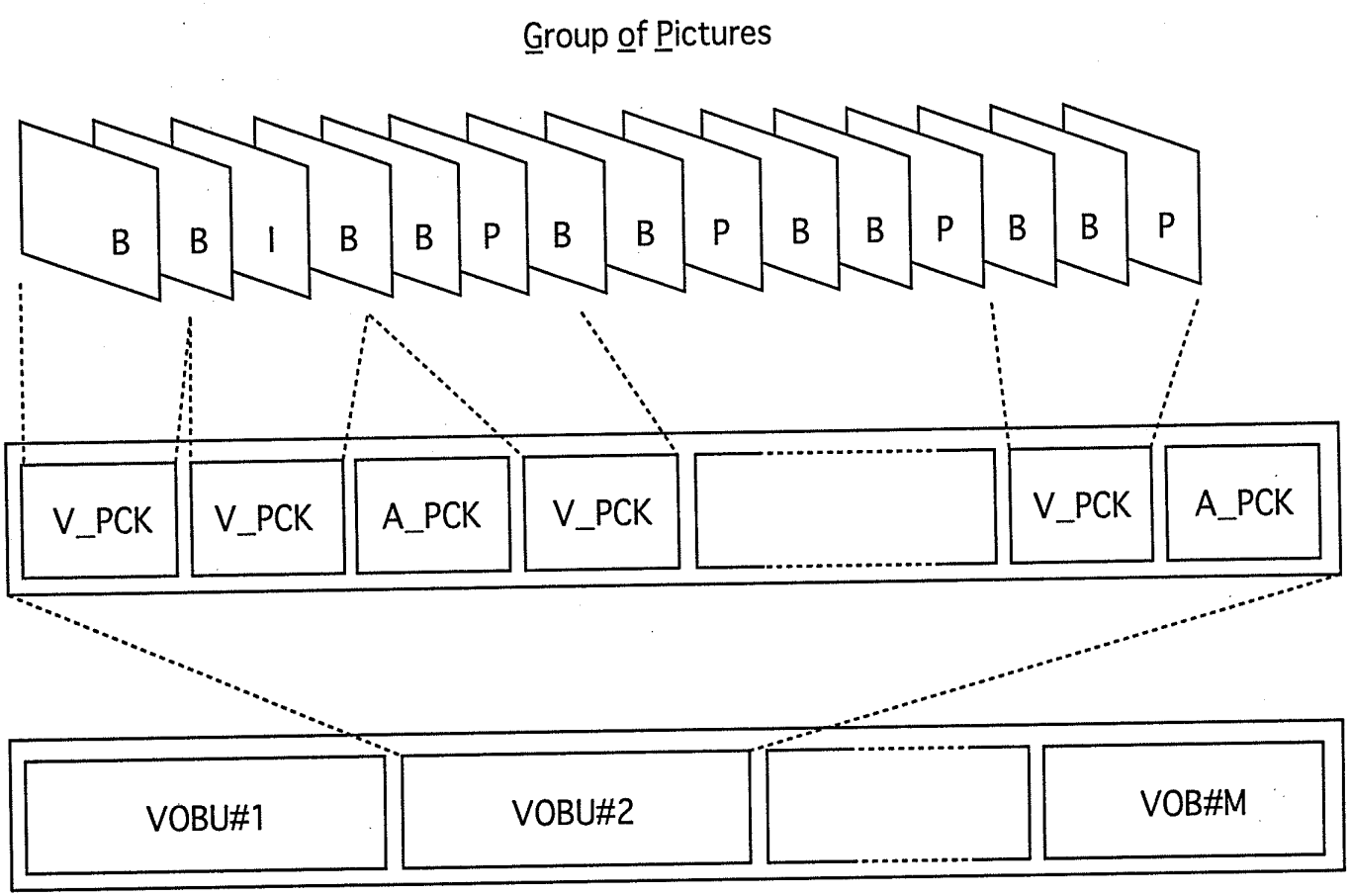

Group of Pictures
B
B
I
B
B
P
B
B
P
B
B
P
B
B
P
V_PCK
V_PCK
A_PCK
V_PCK
V_PCK
A_PCK
VOBU#1
VOBU#2
VOB#M

FIG.10

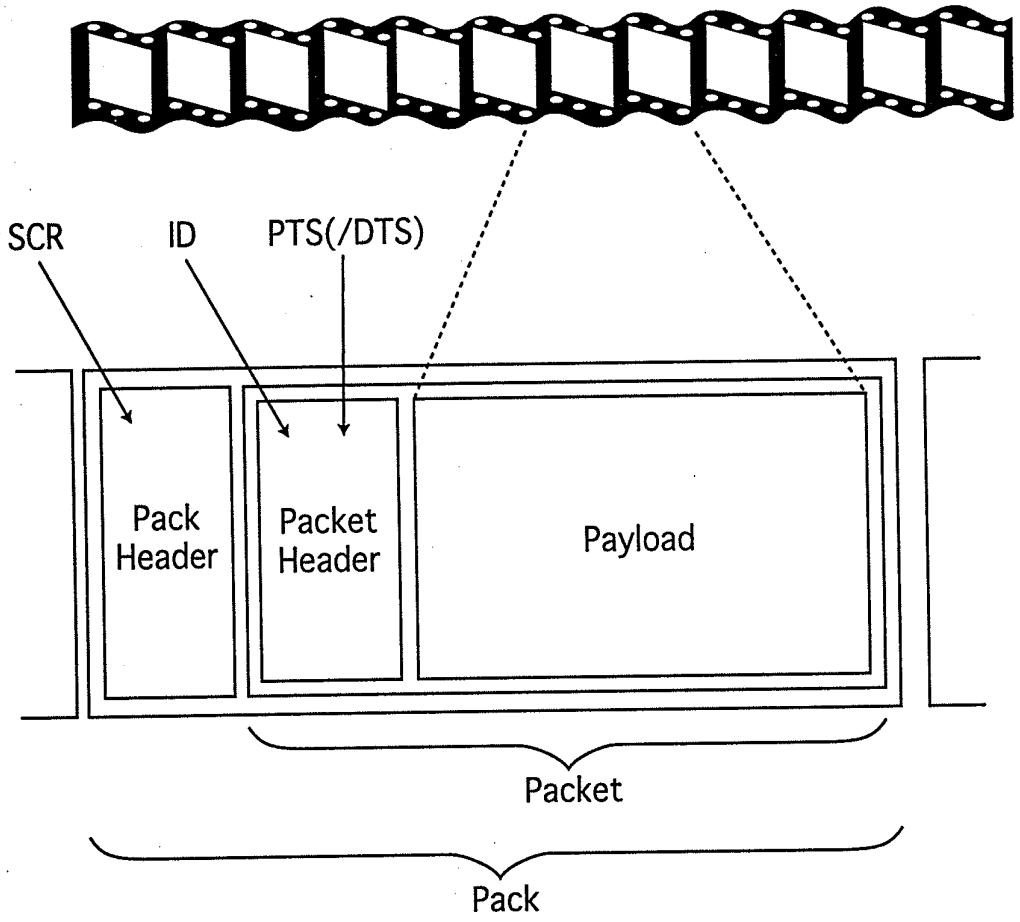

SCR
ID
PTS(/DTS)
Pack Header
Packet Header
Payload
Packet
Pack

FIG.11

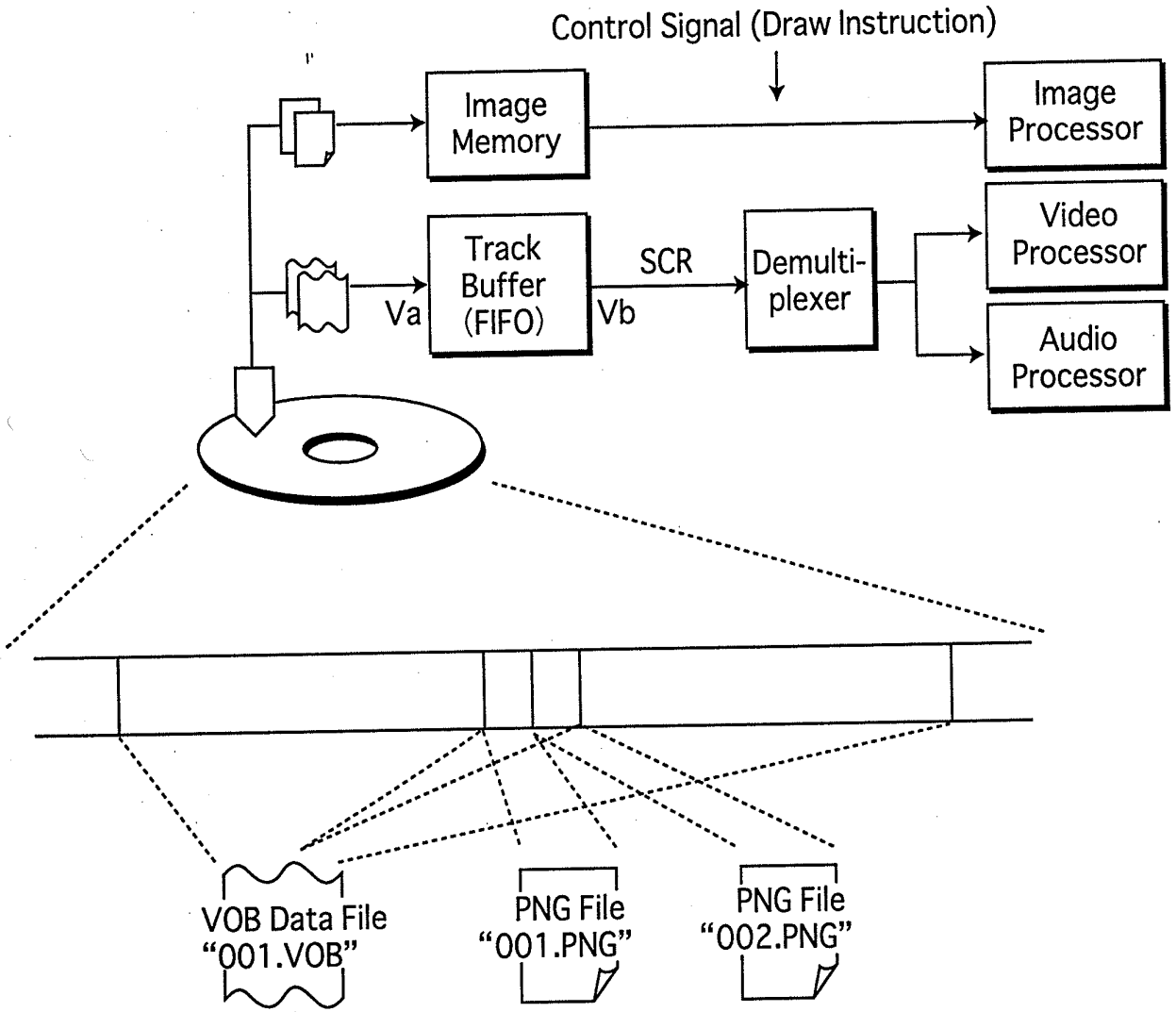

Control Signal (Draw Instruction)
Image Memory
Image Processor
Track Buffer (FIFO)
Va
Vb
SCR
Demulti-plexer
Video Processor
Audio Processor
VOB Data File "001.VOB"
PNG File "001.PNG"
PNG File "002.PNG"

FIG.12

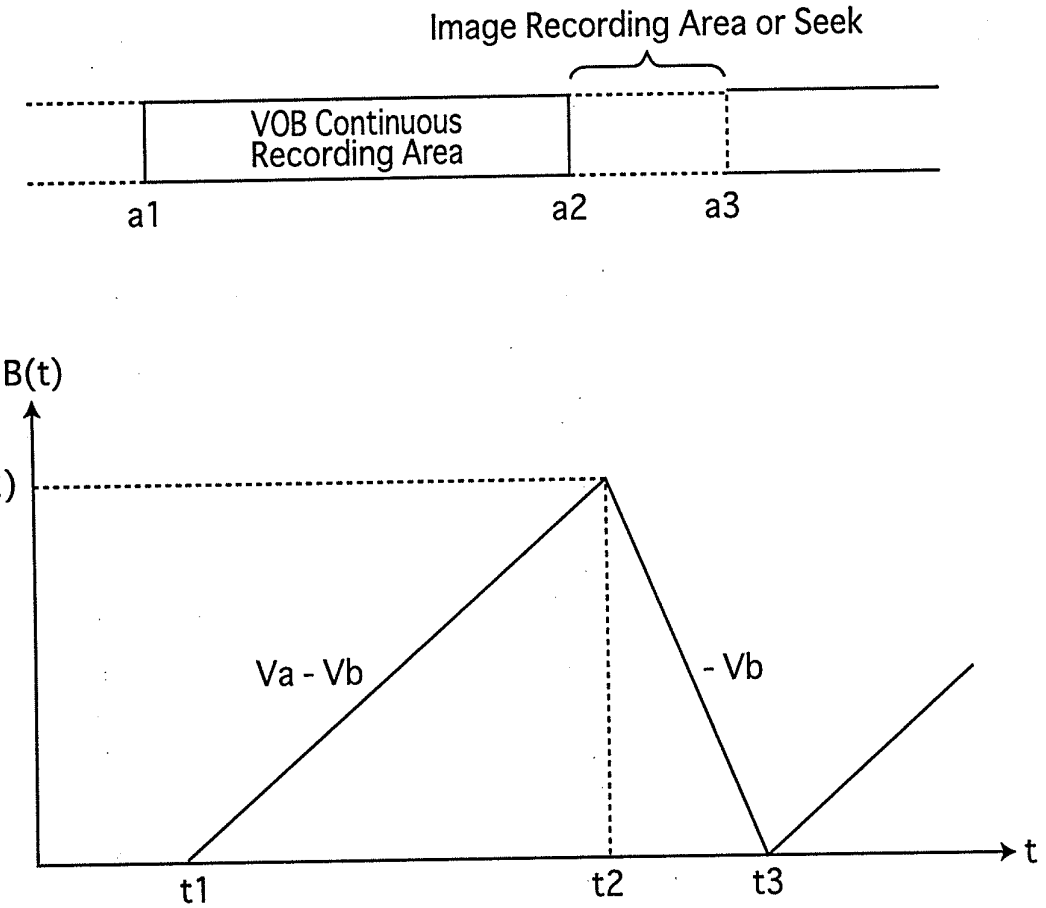


Image Recording Area or Seek
VOB Continuous Recording Area
a1
a2
a3
B(t)
B(t2)
Va - Vb
- Vb
t
t1
t2
t3

FIG.13

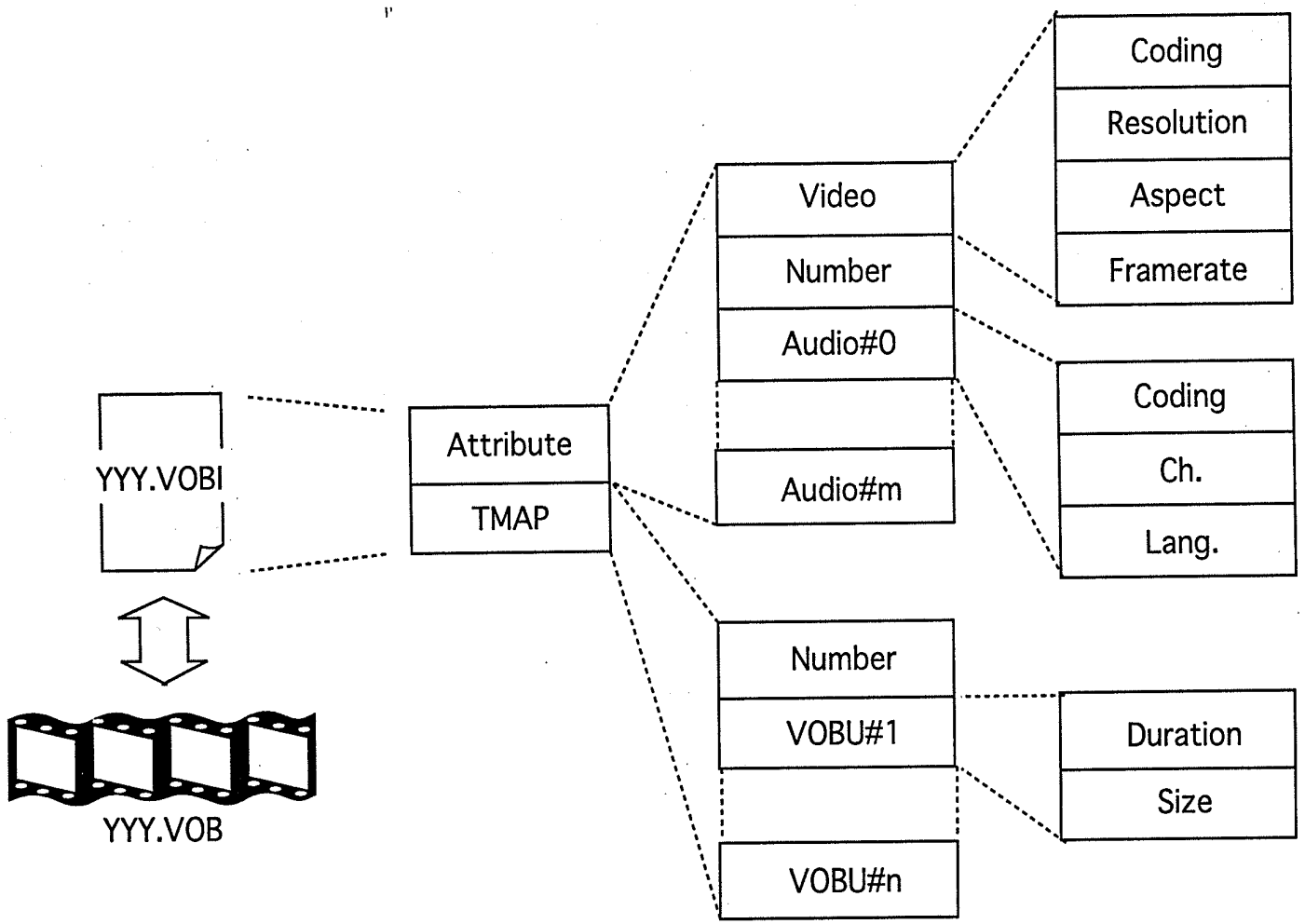

YYY.VOBI
YYY.VOB
Attribute
TMAP
Video
Number
Audio#0
Audio#m
Number
VOBU#1
VOBU#n
Coding
Resolution
Aspect
Framerate
Coding
Ch.
Lang.
Duration
Size

FIG.14

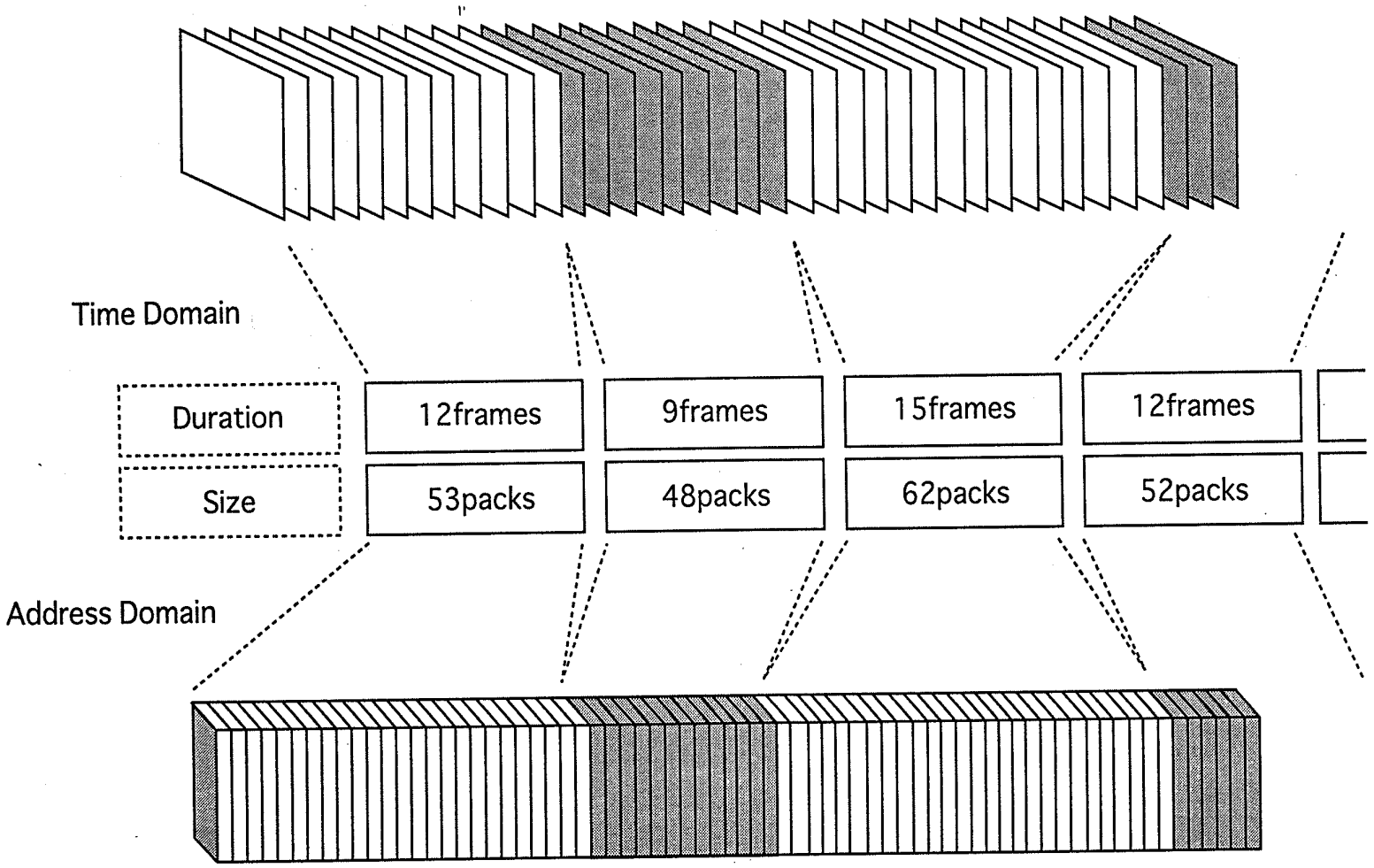

Time Domain
Duration
12frames
9frames
15frames
12frames
Size
53packs
48packs
62packs
52packs
Address Domain

FIG15

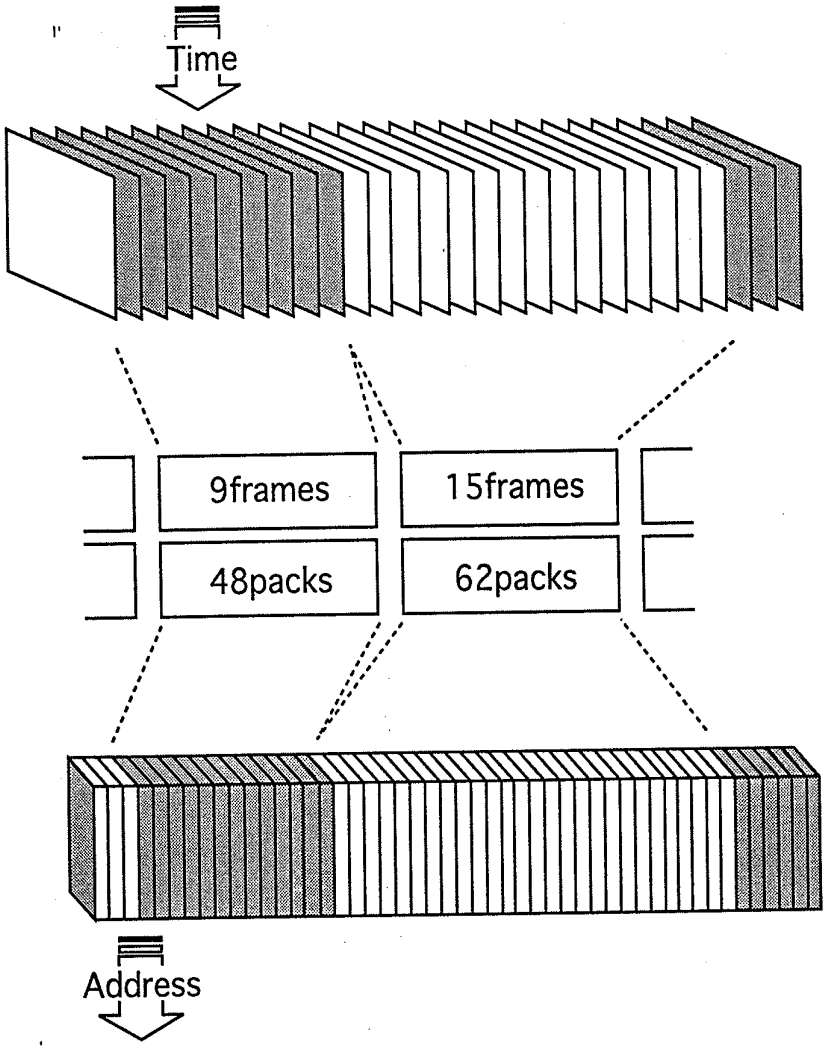


FIG.16

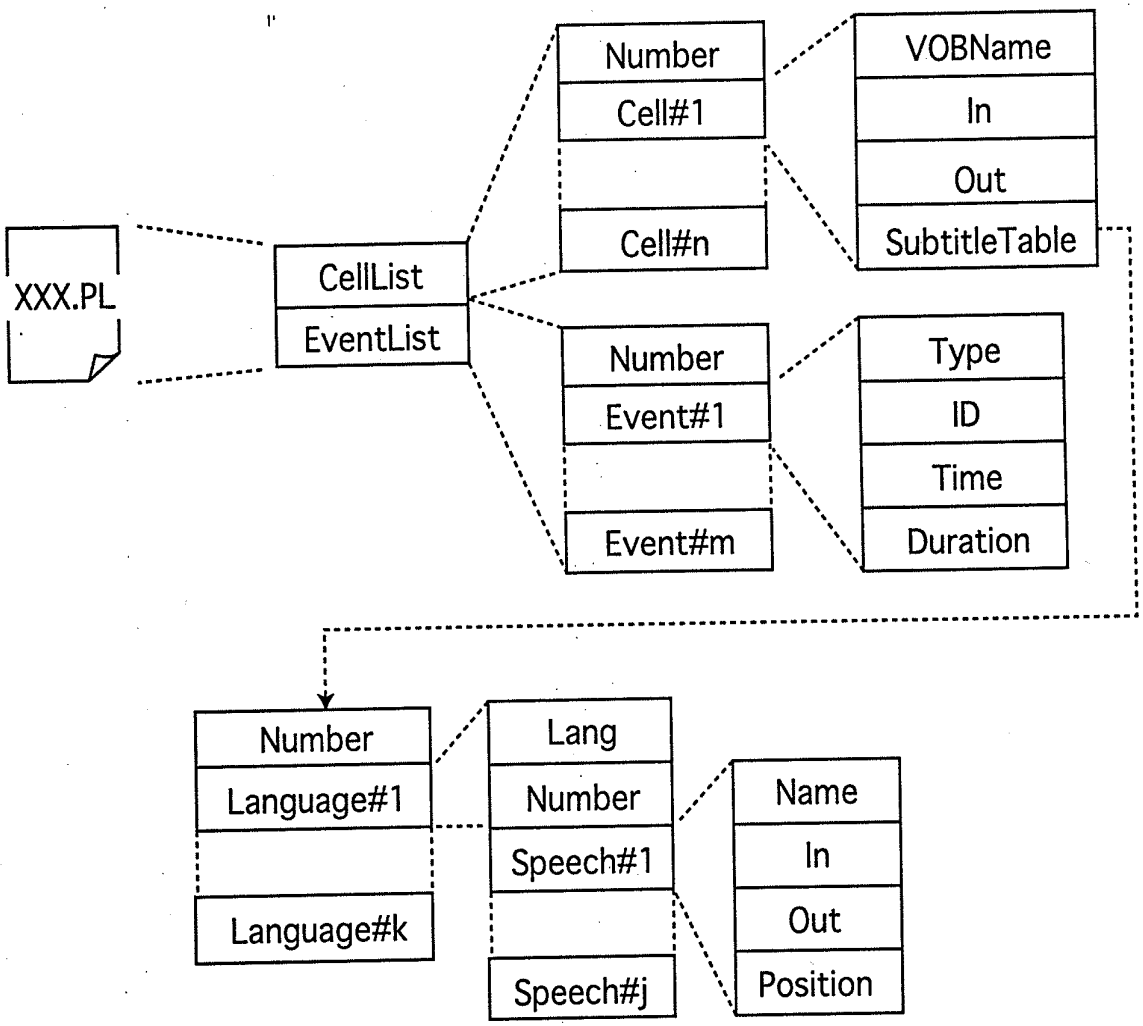

XXX.PL
CellList
EventList
Number
Cell#1
Cell#n
VOBName
In
Out
SubtitleTable
Number
Event#1
Event#m
Type
ID
Time
Duration
Number
Language#1
Language#k
Lang
Number
Speech#1
Speech#j
Name
In
Out
Position

FIG.17

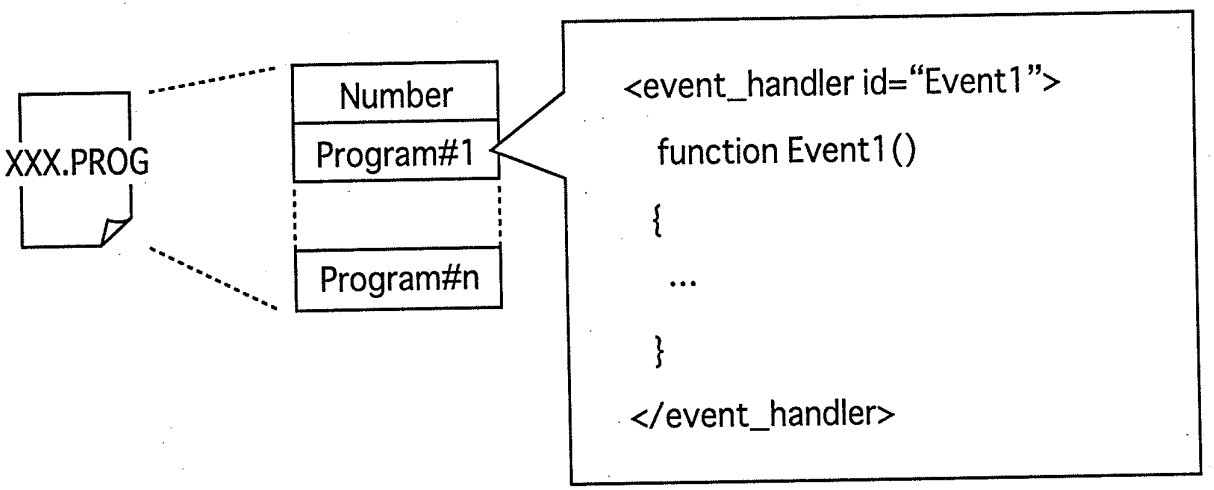

XXX.PROG
Number
Program#1
Program#n
<event_handler id="Event1">
function Event1()
{
...
}
</event_handler>

FIG.18

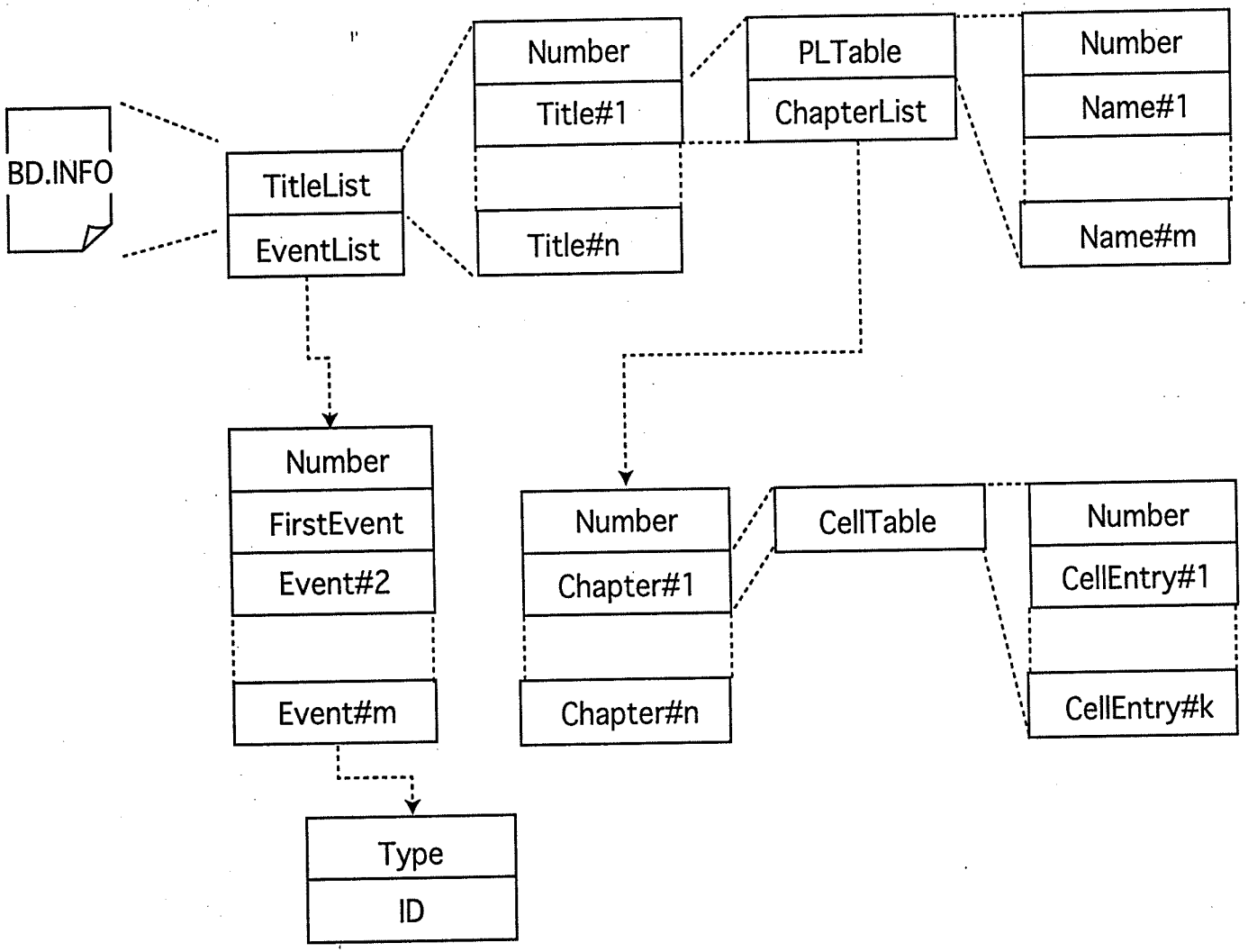

BD.INFO
TitleList
EventList
Number
Title#1
Title#n
PLTable
ChapterList
Number
Name#1
Name#m
Number
FirstEvent
Event#2
Event#m
Number
Chapter#1
Chapter#n
CellTable
Number
CellEntry#1
CellEntry#k
Type
ID

FIG.19

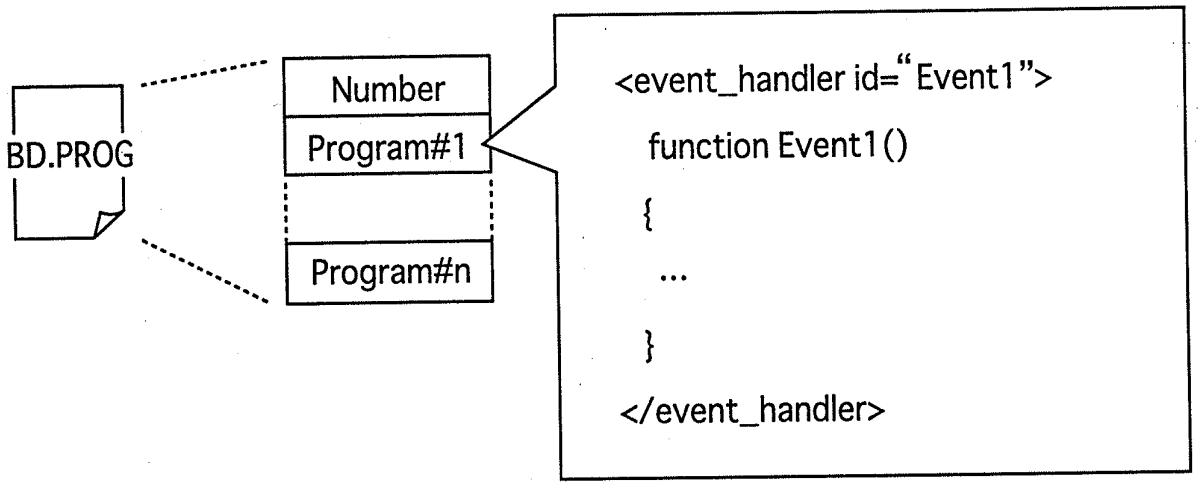

BD.PROG
Number
Program#1
Program#n
<event_handler id="Event1">
function Event1()
{
...
}
</event_handler>

FIG.20

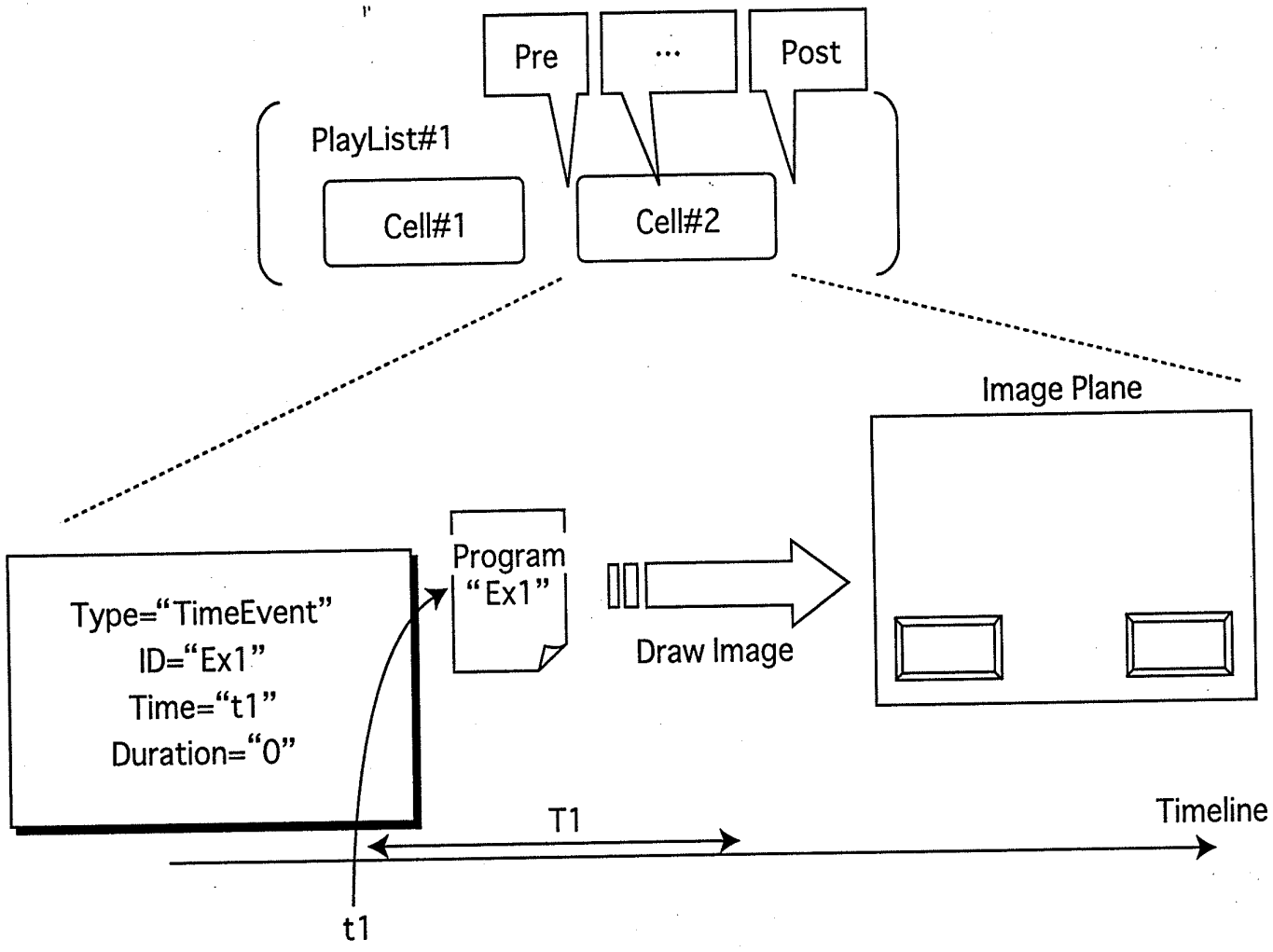

Pre
...
Post
PlayList#1
Cell#1
Cell#2
Image Plane
Type="TimeEvent"
ID="Ex1"
Time="t1"
Duration="0"
Program
"Ex1"
Draw Image
T1
Timeline
t1

FIG.21

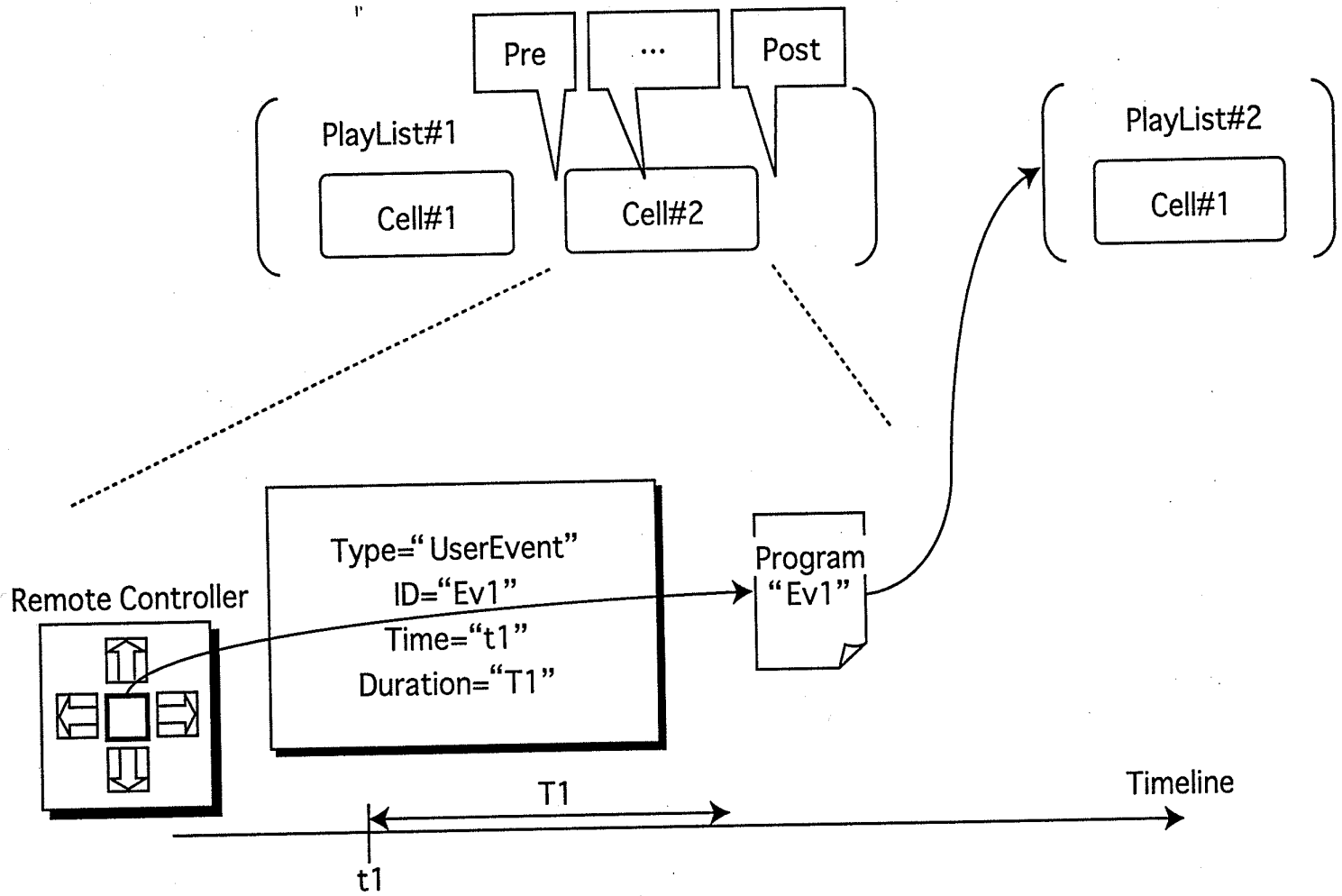

Pre
...
Post
PlayList#1
Cell#1
Cell#2
PlayList#2
Cell#1
Type="UserEvent"
ID="Ev1"
Time="t1"
Duration="T1"
Remote Controller
Program
"Ev1"
T1
Timeline
t1

FIG.22

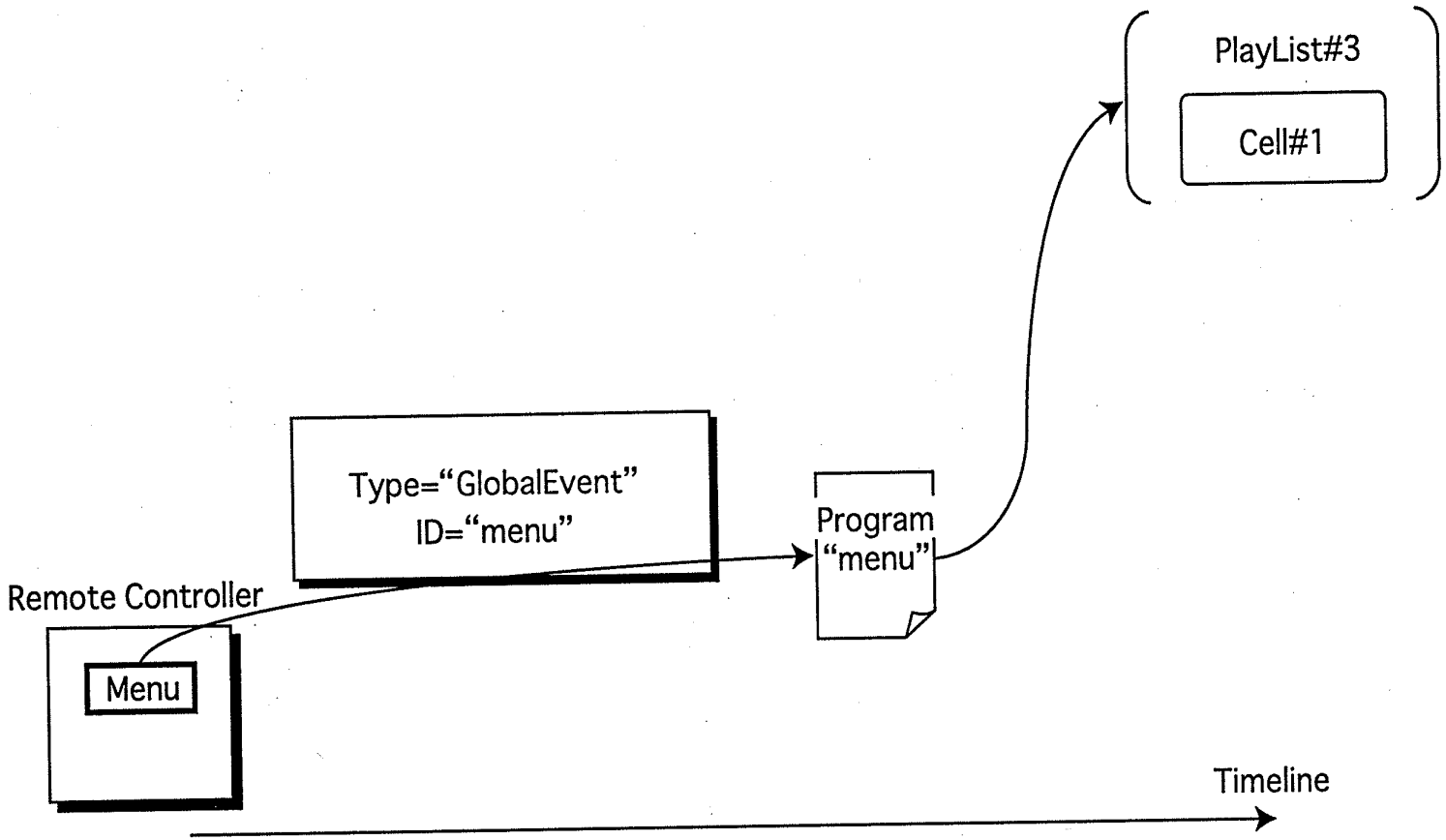

PlayList#3
Cell#1
Type="GlobalEvent"
ID="menu"
Program
"menu"
Remote Controller
Menu
Timeline

FIG.23

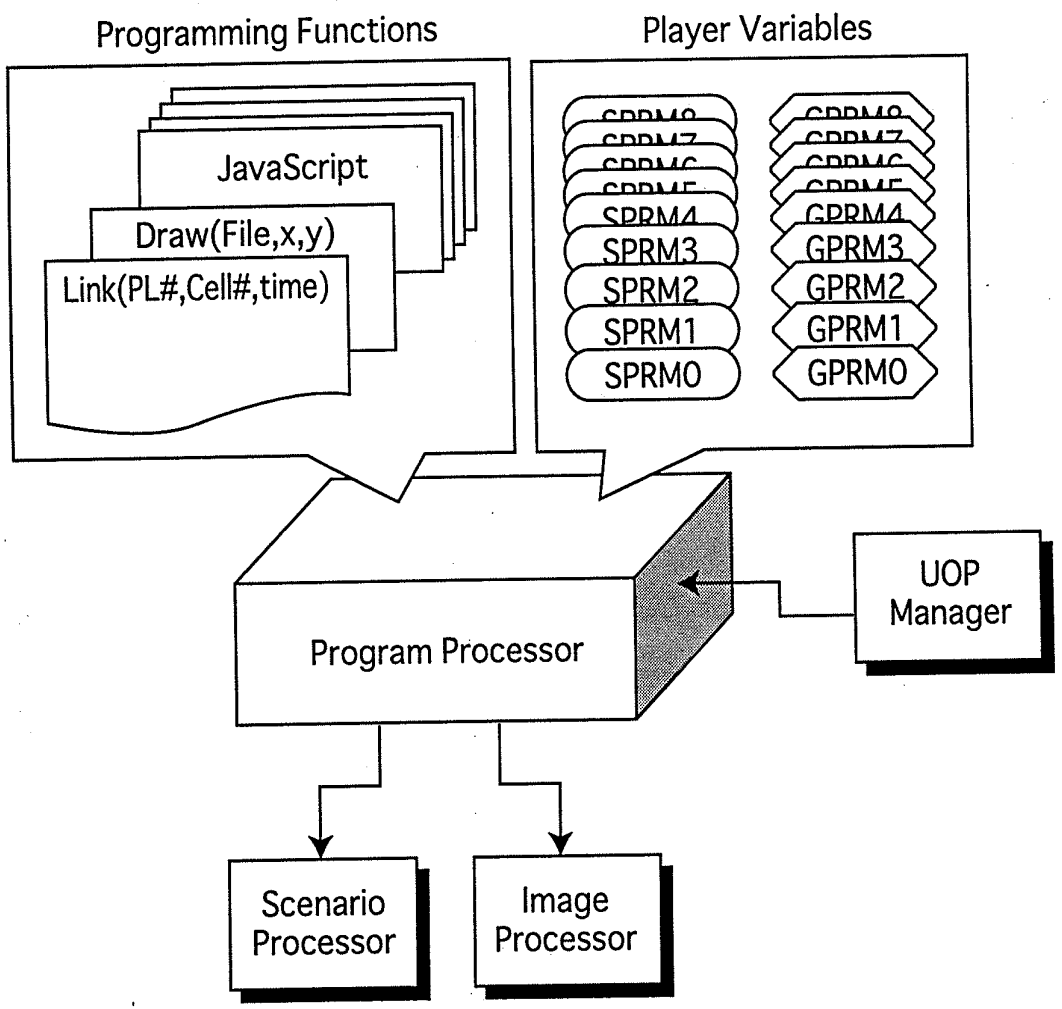

Programming Functions
Player Variables
JavaScript
Draw(File,x,y)
Link(PL#,Cell#,time)
SPRM3
SPRM2
SPRM1
SPRM0
GPRM4
GPRM3
GPRM2
GPRM1
GPRM0
Program Processor
UOP Manager
Scenario Processor
Image Processor

FIG.24

Player Variables (System Parameters)

0	Language Code
1	Audio stream number
2	Subtitle stream number
3	Angle number
4	Title number
5	Chapter number
6	Program number
7	Cell number
8	Key name
9	Navigation timer
10	Current playback time

11	Player audio mixing mode for Karaoke
12	Country code for parental management
13	Parental level
14	Player configuration for Video
15	Player configuration for Audio
16	Language code for AST
17	Language code ext. for AST
18	Language code for STST
19	Language coded ext. for STST
20	Player region code
21	reserved

22	reserved
23	Player status
24	reserved
25	reserved
26	reserved
27	reserved
28	reserved
29	reserved
30	reserved
31	reserved
32	reserved

FIG.25

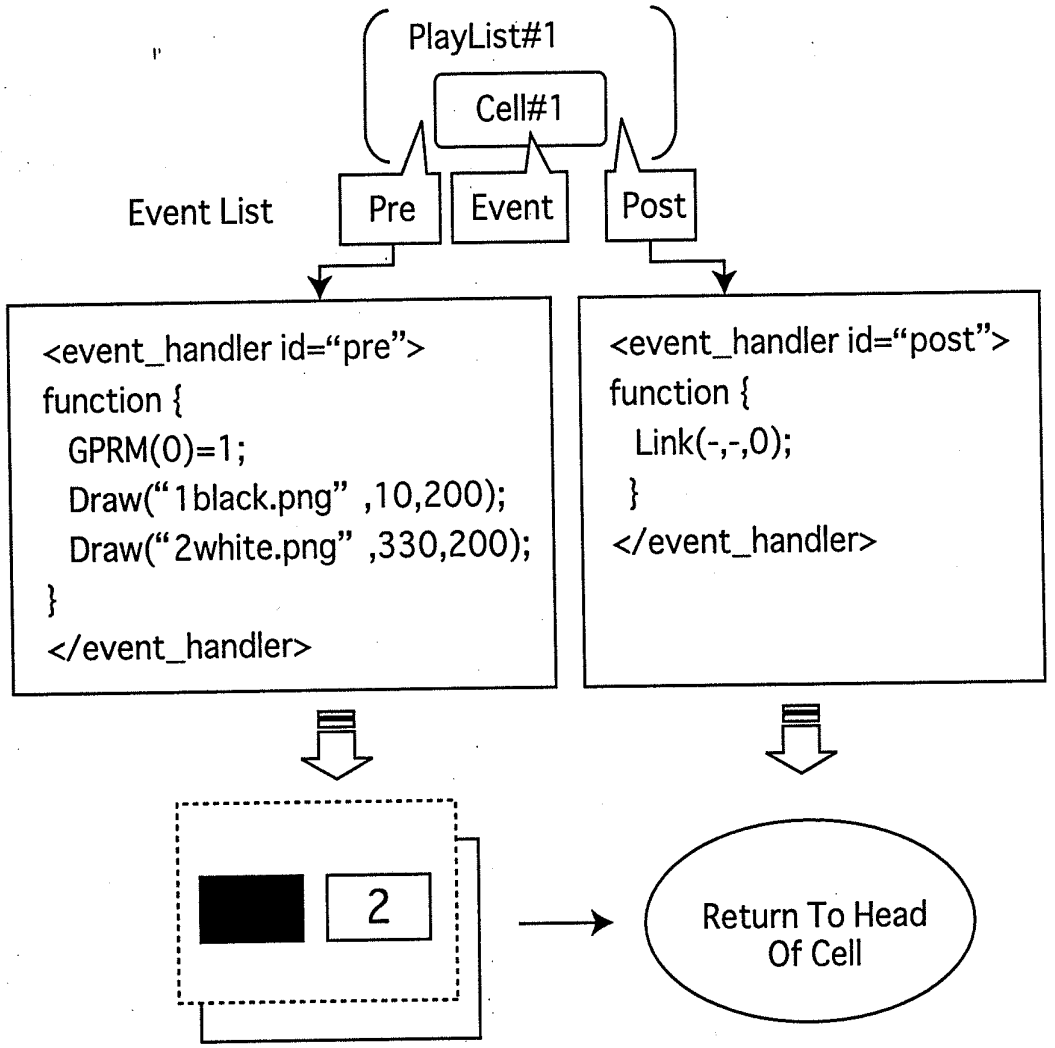

PlayList#1
Cell#1
Event List
Pre
Event
Post
<event_handler id="pre">
function {
GPRM(0)=1;
Draw("1black.png" ,10,200);
Draw("2white.png" ,330,200);
}
</event_handler>
<event_handler id="post">
function {
Link(-,-,0);
}
</event_handler>
2
Return To Head Of Cell

FIG.26

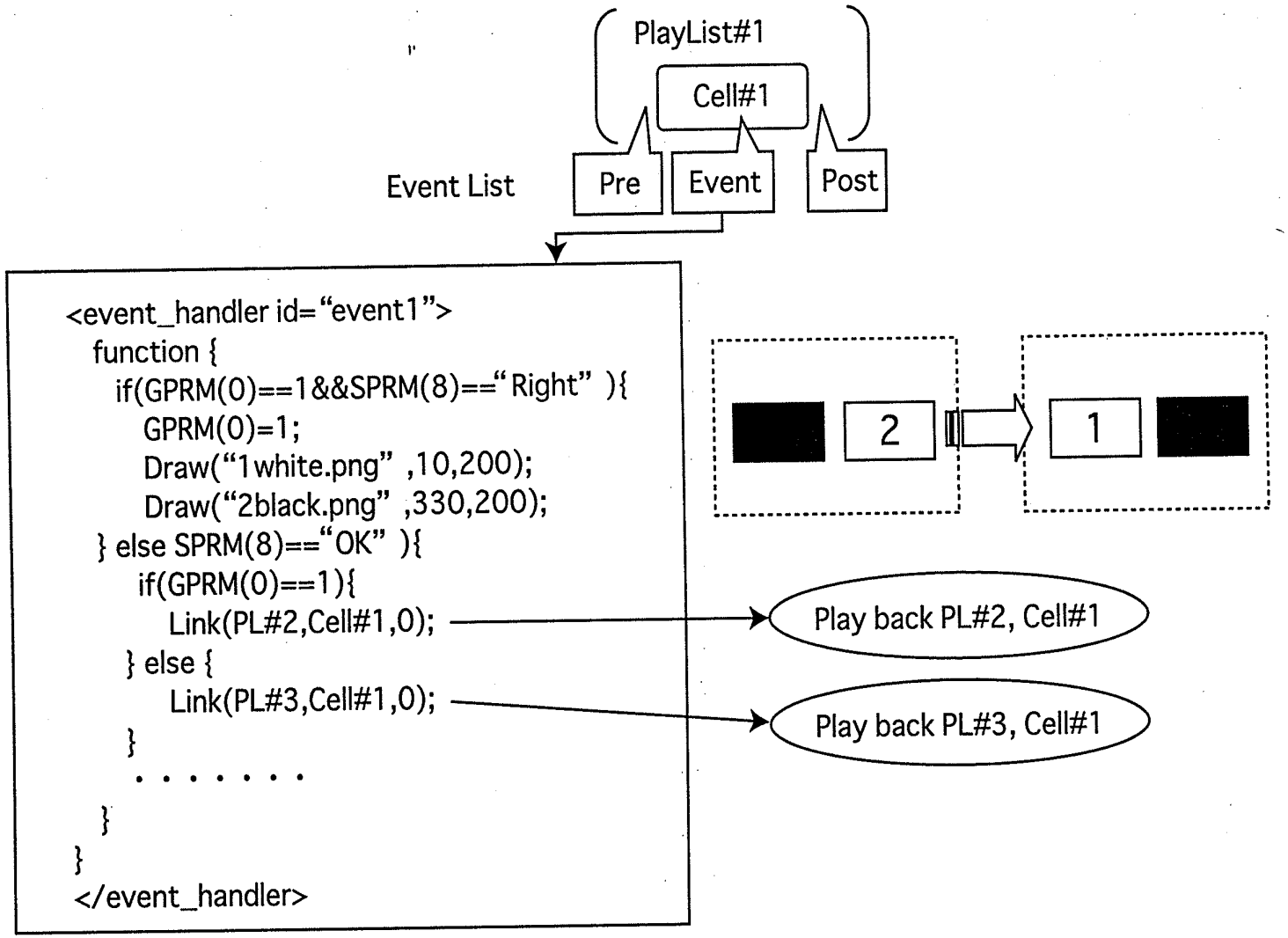

PlayList#1
Cell#1
Event List
Pre
Event
Post
<event_handler id="event1">
function {
if(GPRM(0)==1&&SPRM(8)=="Right"){
GPRM(0)=1;
Draw("1white.png" ,10,200);
Draw("2black.png" ,330,200);
} else SPRM(8)=="OK"){
if(GPRM(0)==1){
Link(PL#2,Cell#1,0);
} else {
Link(PL#3,Cell#1,0);
}
. . . . . .
}
}
</event_handler>
2
1
Play back PL#2, Cell#1
Play back PL#3, Cell#1

FIG.27

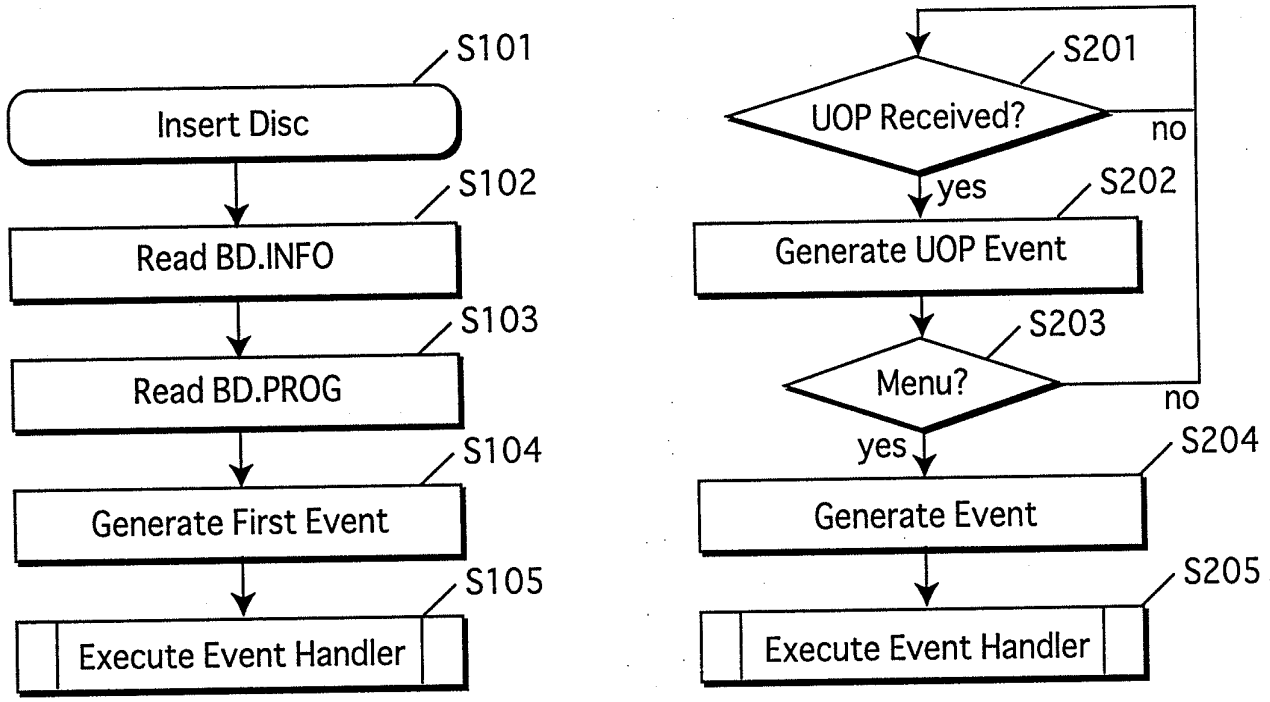

S101
Insert Disc
S102
Read BD.INFO
S103
Read BD.PROG
S104
Generate First Event
S105
Execute Event Handler
S201
UOP Received?
no
yes
S202
Generate UOP Event
S203
Menu?
no
yes
S204
Generate Event
S205
Execute Event Handler

FIG.28

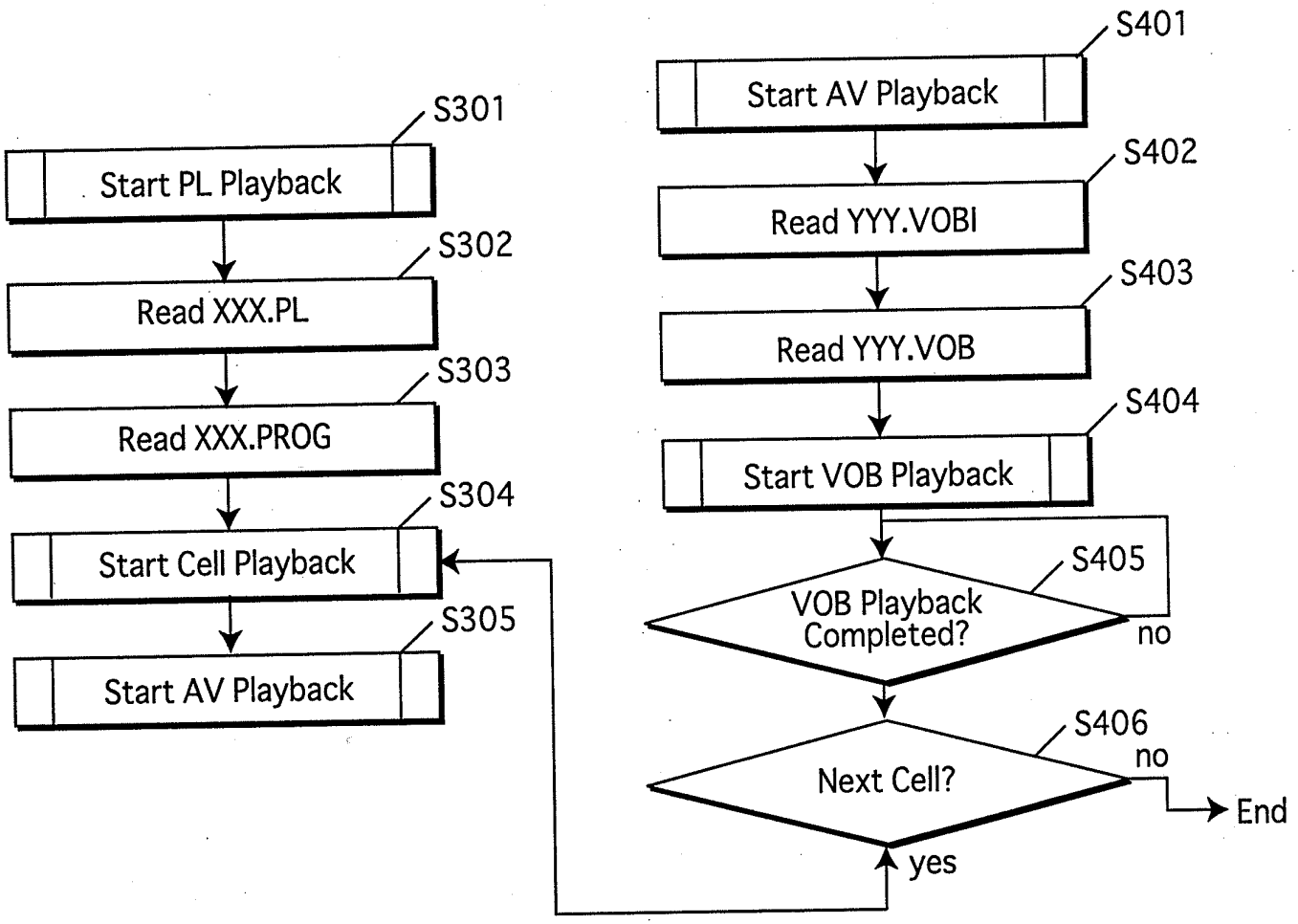

S301
Start PL Playback
S302
Read XXX.PL
S303
Read XXX.PROG
S304
Start Cell Playback
S305
Start AV Playback
S401
Start AV Playback
S402
Read YYY.VOBI
S403
Read YYY.VOB
S404
Start VOB Playback
S405
VOB Playback Completed?
no
S406
Next Cell?
no
End
yes

FIG.29

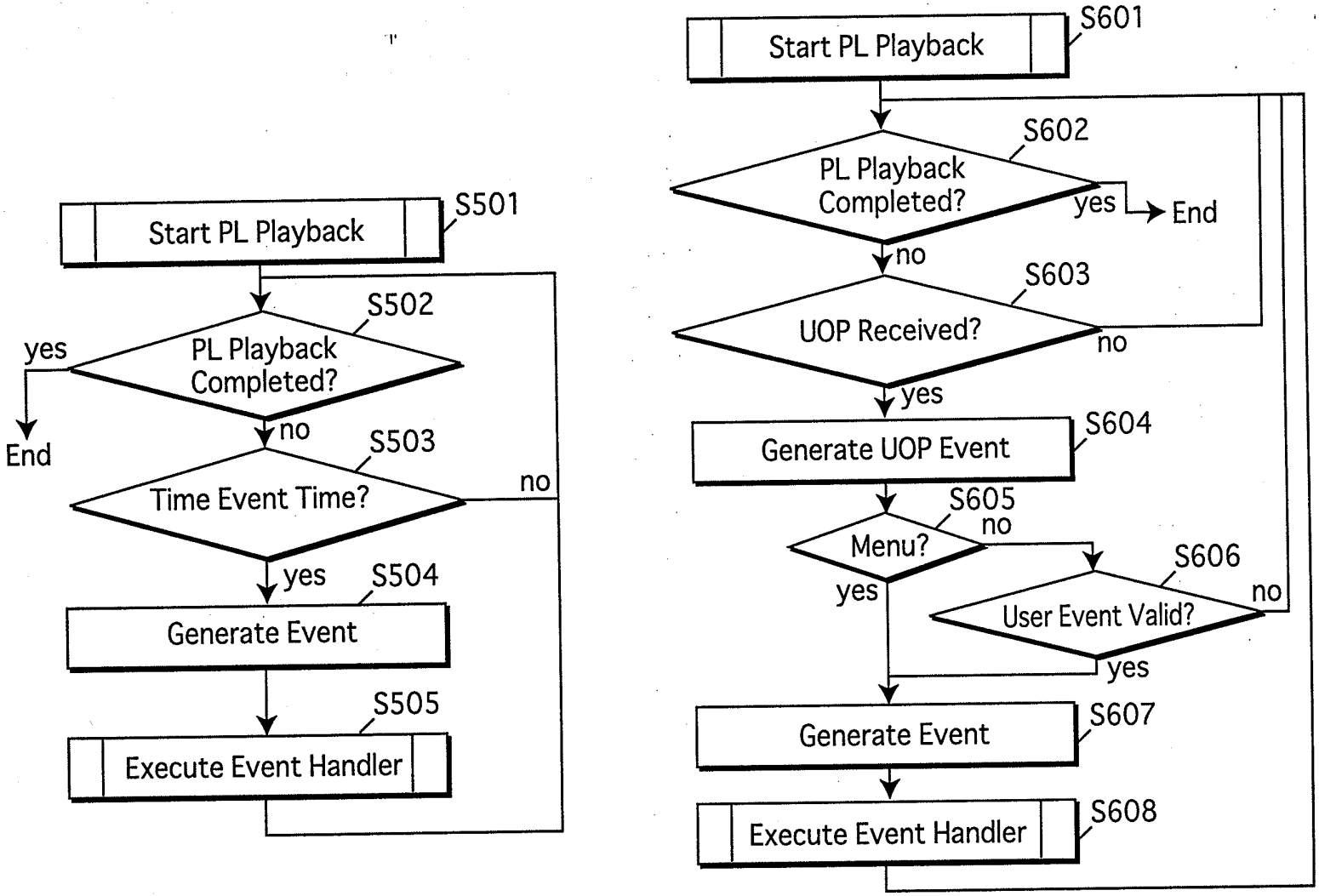

Start PL Playback
S501
S502
PL Playback Completed?
yes
End
no
S503
Time Event Time?
no
yes
S504
Generate Event
S505
Execute Event Handler
Start PL Playback
S601
S602
PL Playback Completed?
yes
End
no
S603
UOP Received?
no
yes
S604
Generate UOP Event
S605
Menu?
no
yes
S606
User Event Valid?
no
yes
S607
Generate Event
S608
Execute Event Handler

FIG.30

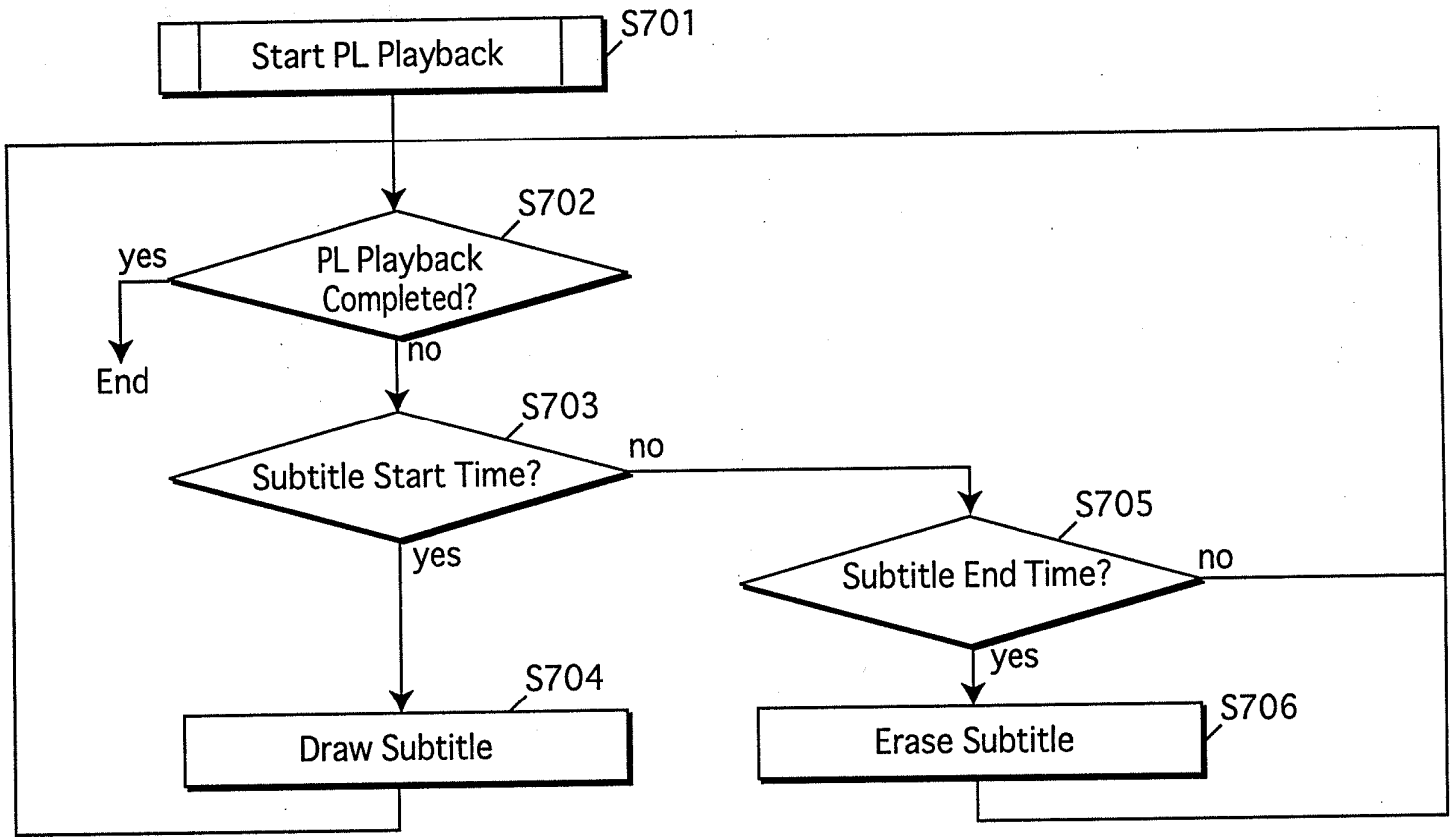

Start PL Playback
S701
S702
PL Playback Completed?
yes
no
End
S703
Subtitle Start Time?
no
yes
S704
Draw Subtitle
S705
Subtitle End Time?
no
yes
S706
Erase Subtitle

FIG.31

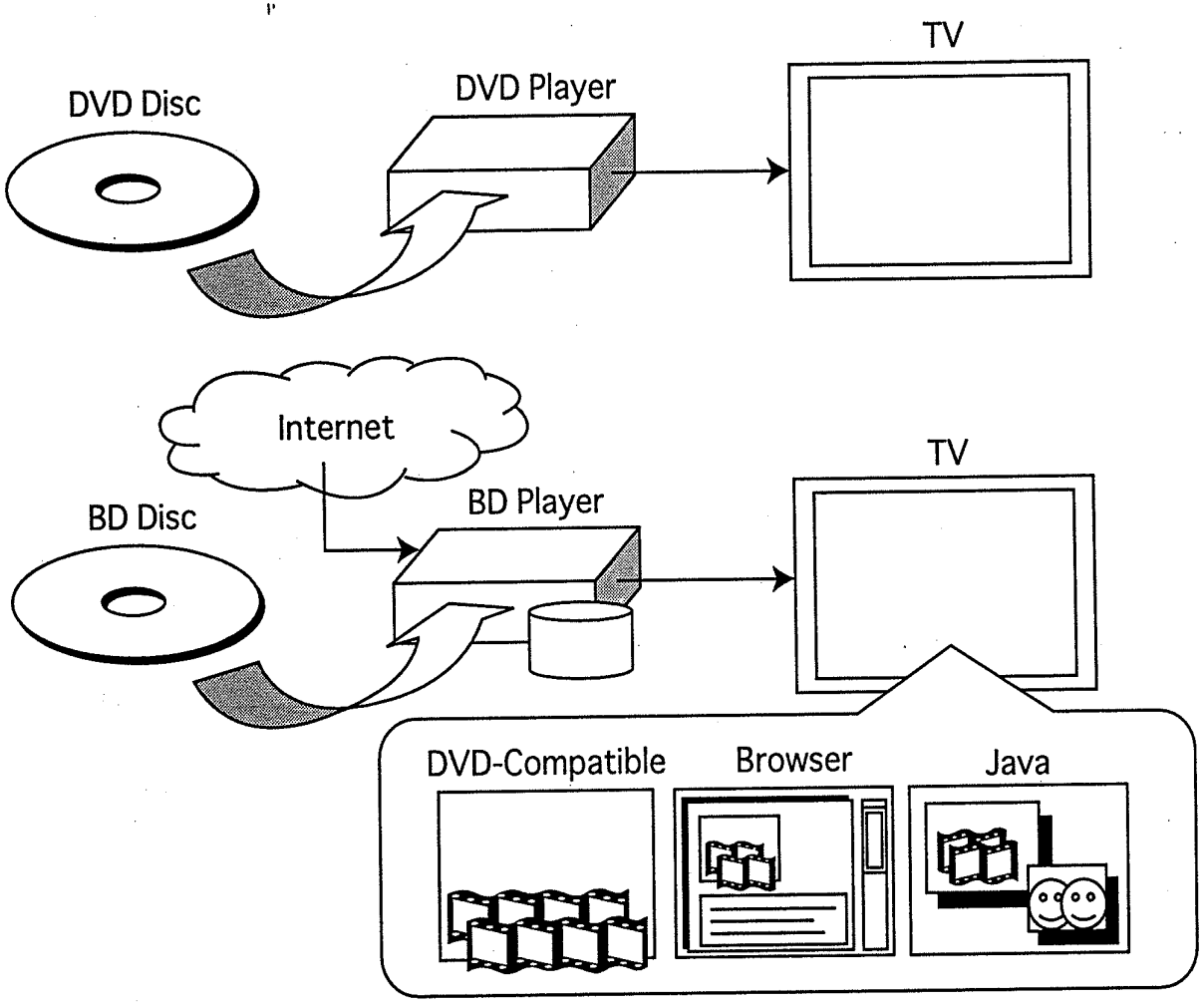

TV
DVD Disc
DVD Player
Internet
TV
BD Disc
BD Player
DVD-Compatible
Browser
Java

FIG.32

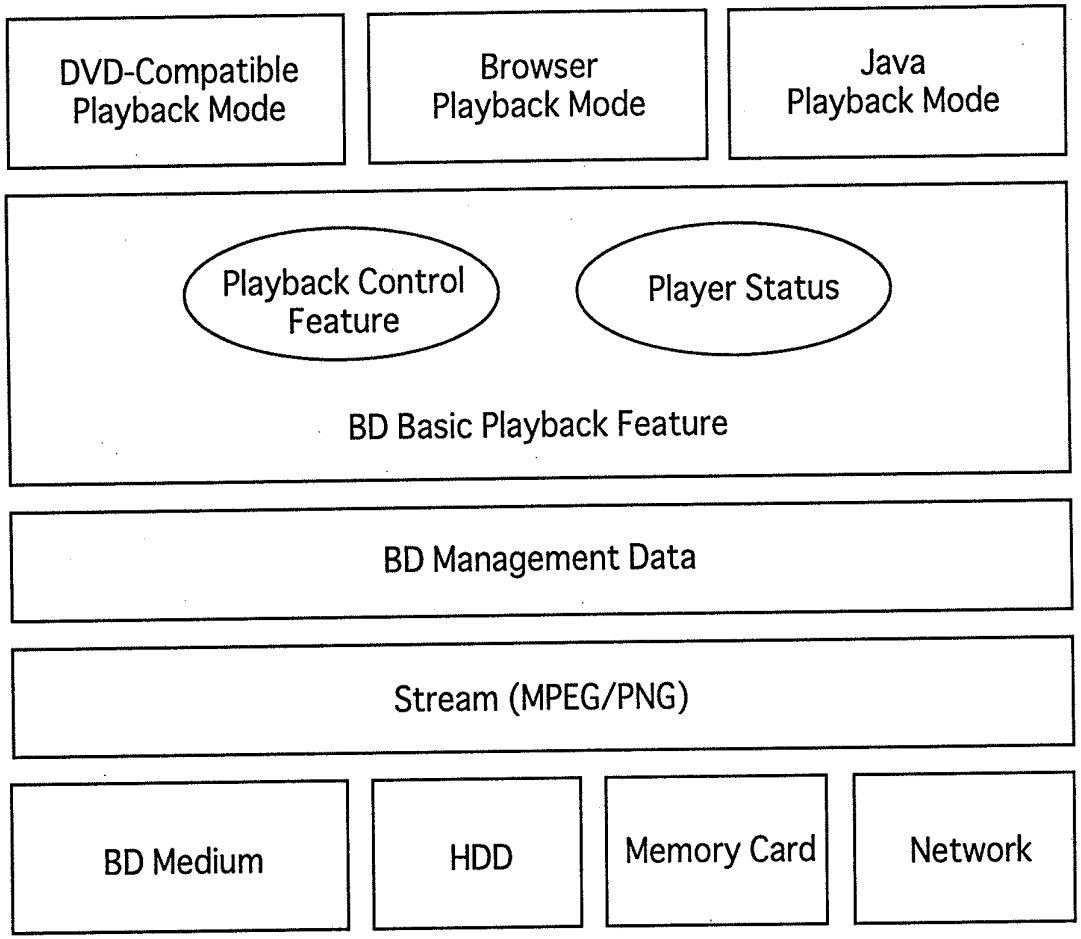

DVD-Compatible Playback Mode
Browser Playback Mode
Java Playback Mode
Playback Control Feature
Player Status
BD Basic Playback Feature
BD Management Data
Stream (MPEG/PNG)
BD Medium
HDD
Memory Card
Network

FIG.33

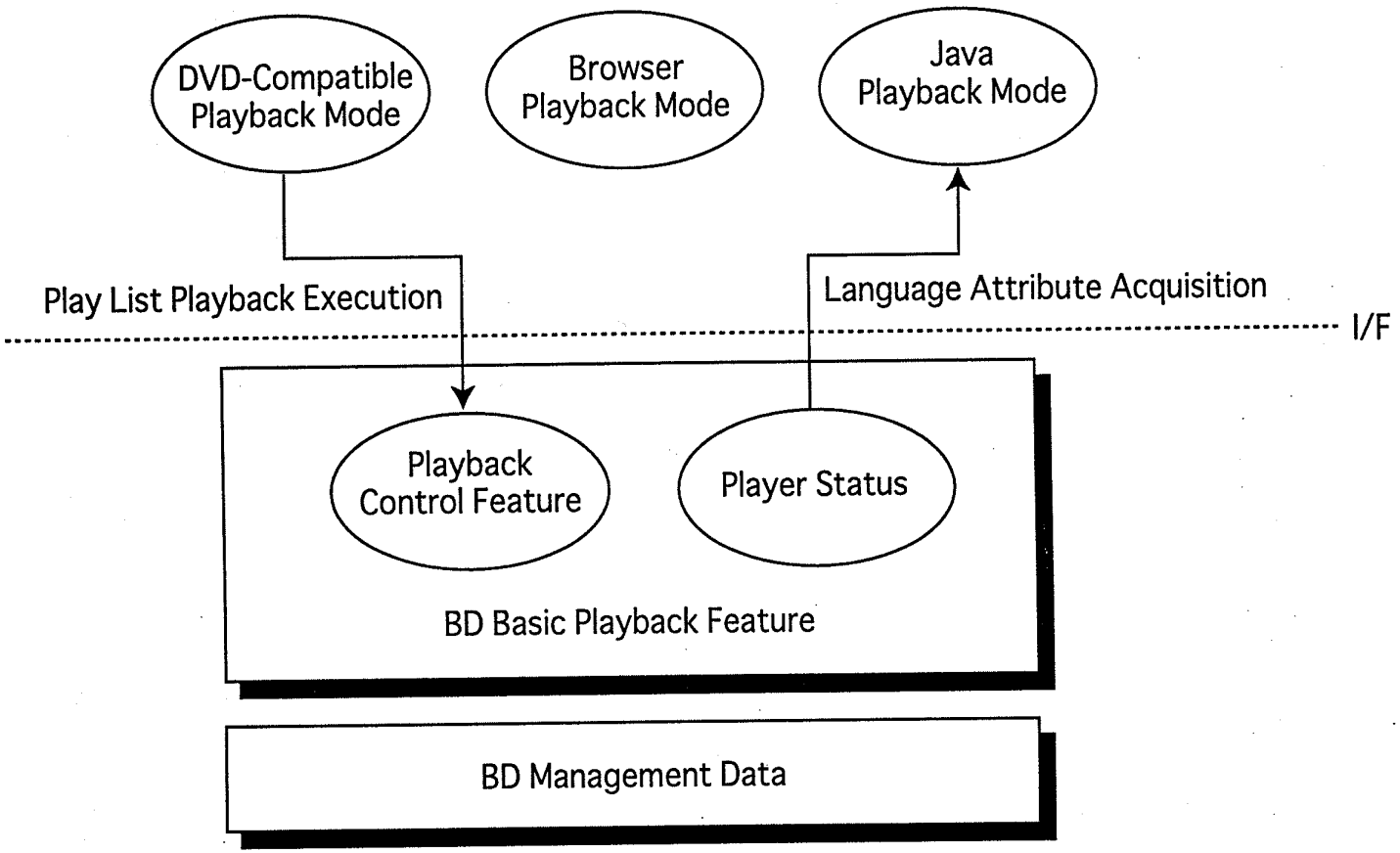

DVD-Compatible Playback Mode
Browser Playback Mode
Java Playback Mode
Play List Playback Execution
Language Attribute Acquisition
I/F
Playback Control Feature
Player Status
BD Basic Playback Feature
BD Management Data

FIG.34

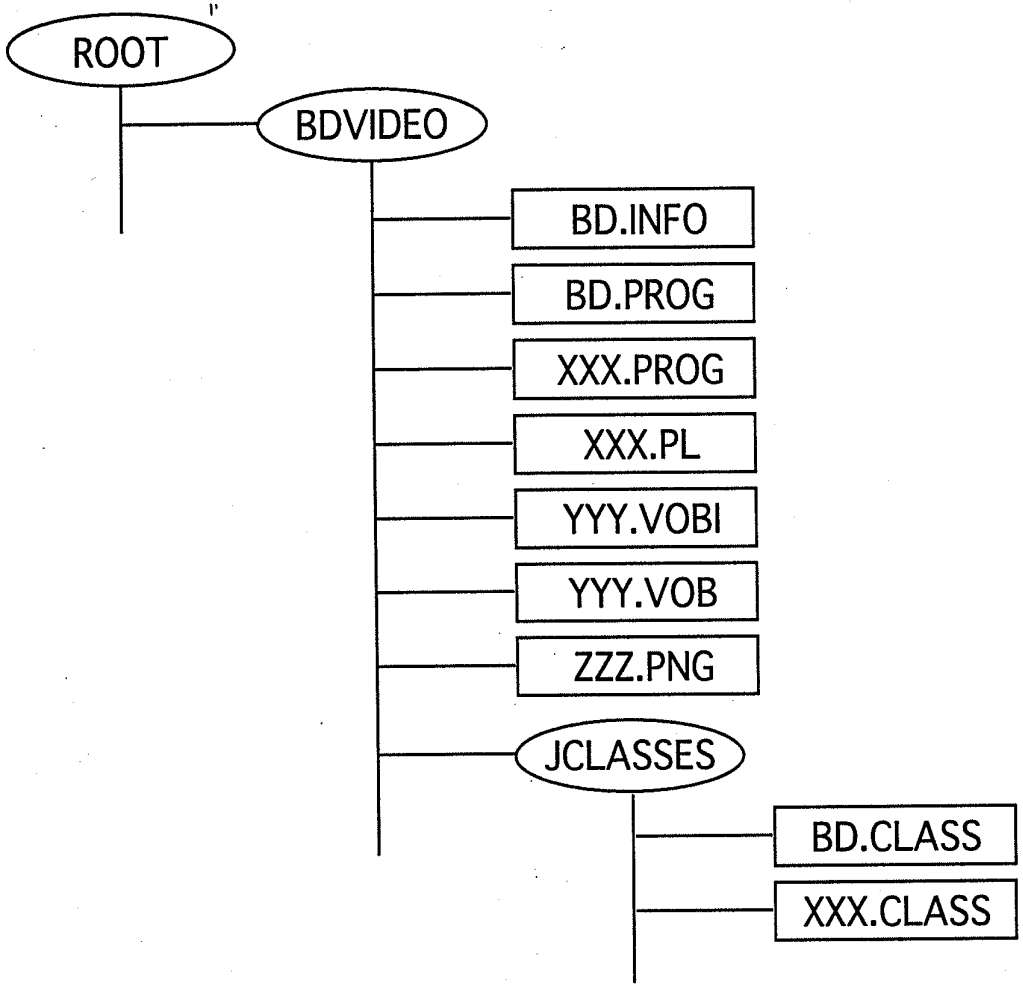

ROOT
BDVIDEO
BD.INFO
BD.PROG
XXX.PROG
XXX.PL
YYY.VOBI
YYY.VOB
ZZZ.PNG
JCLASSES
BD.CLASS
XXX.CLASS

FIG.35

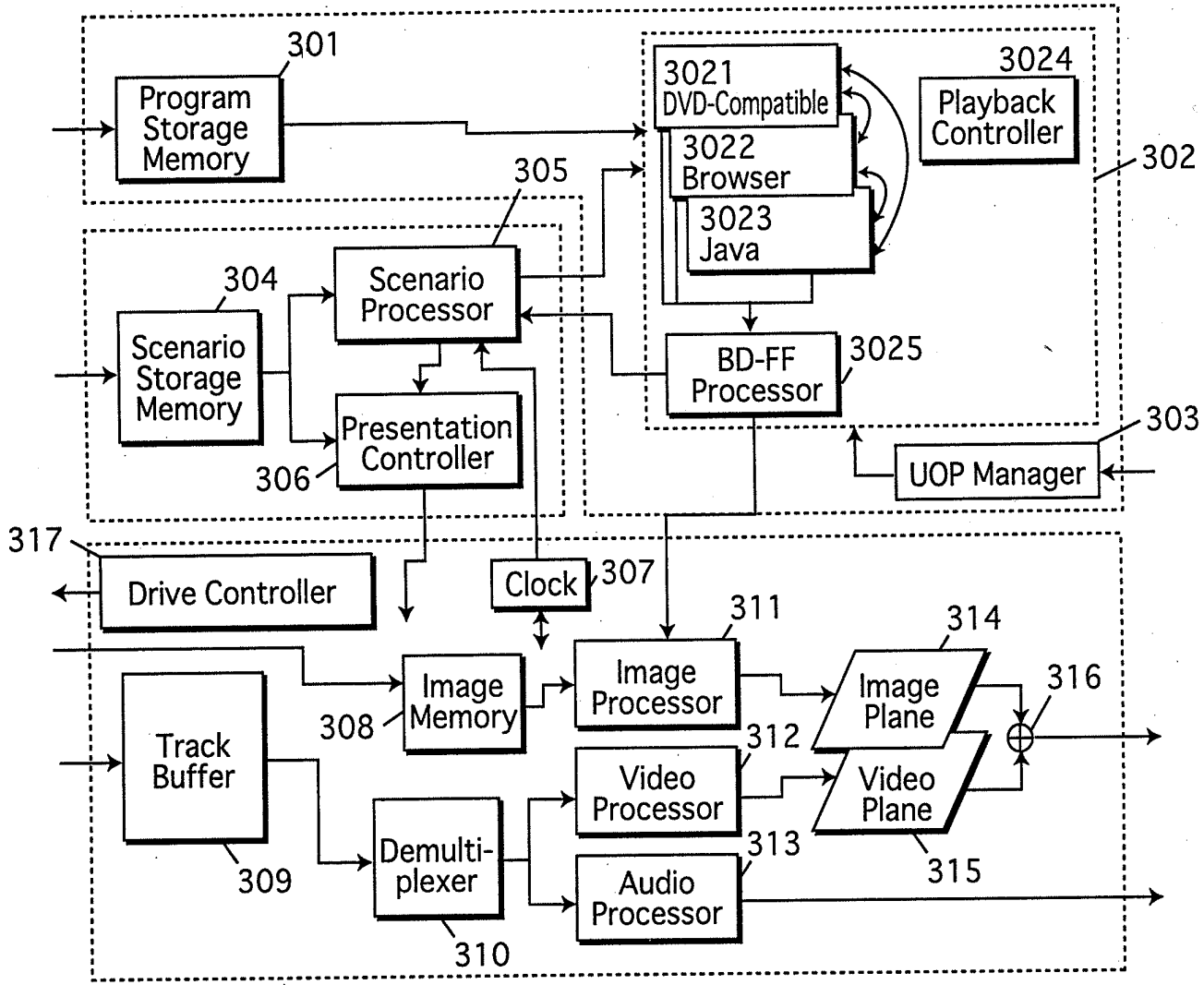

301
Program Storage Memory
3021 DVD-Compatible
3022 Browser
3023 Java
3024 Playback Controller
302
305
304
Scenario Processor
Scenario Storage Memory
BD-FF Processor
3025
Presentation Controller
306
303
UOP Manager
317
Drive Controller
Clock
307
311
314
316
Image Processor
Image Plane
Image Memory
308
312
Track Buffer
Video Processor
Video Plane
313
Demulti-plexer
Audio Processor
309
315
310

FIG.36

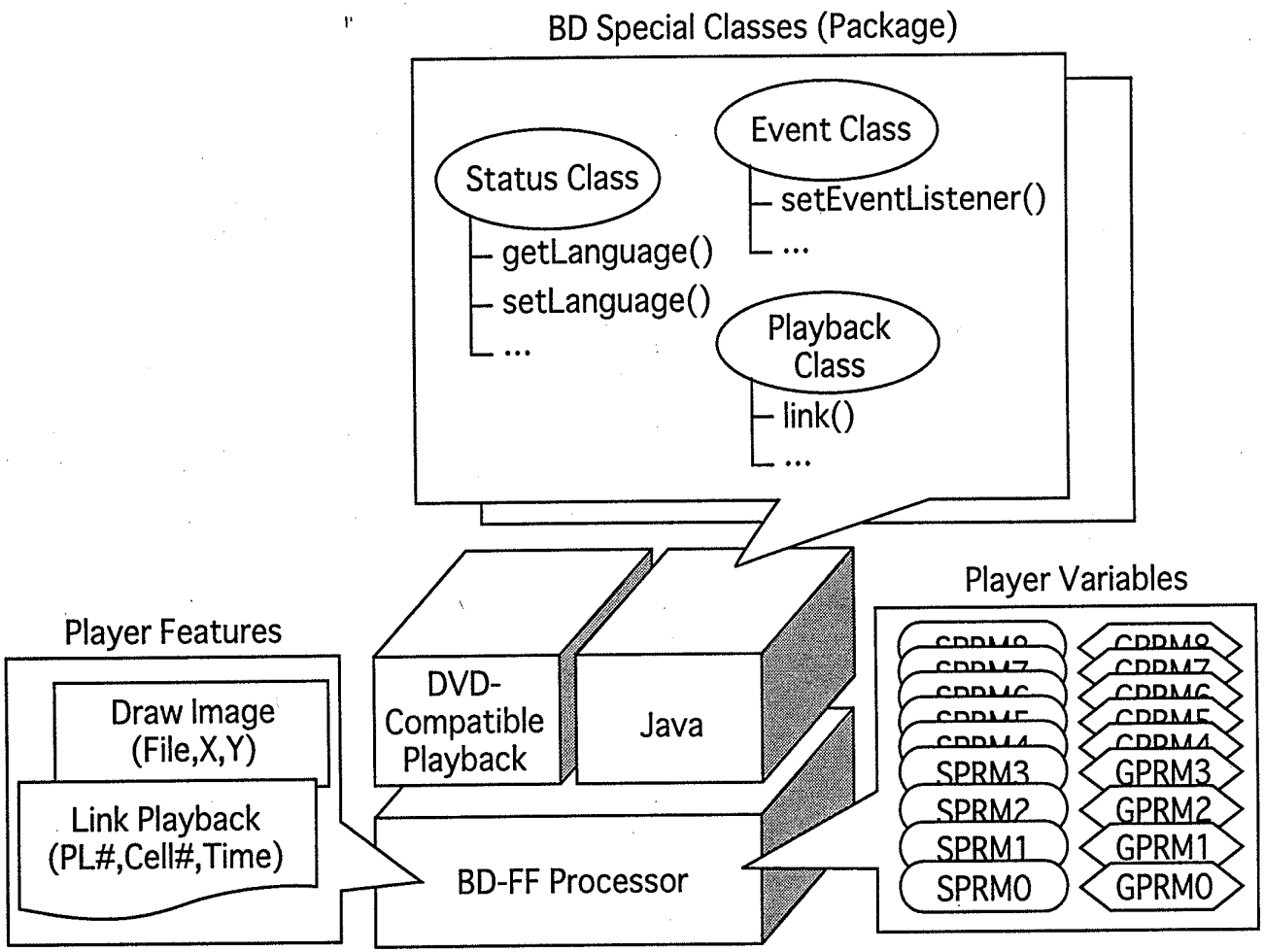

BD Special Classes (Package)
Status Class
getLanguage()
setLanguage()
...
Event Class
setEventListener()
...
Playback Class
link()
...
Player Features
Draw Image (File,X,Y)
Link Playback (PL#,Cell#,Time)
DVD-Compatible Playback
Java
BD-FF Processor
Player Variables
SPRM3
SPRM2
SPRM1
SPRM0
GPRM3
GPRM2
GPRM1
GPRM0

FIG.37

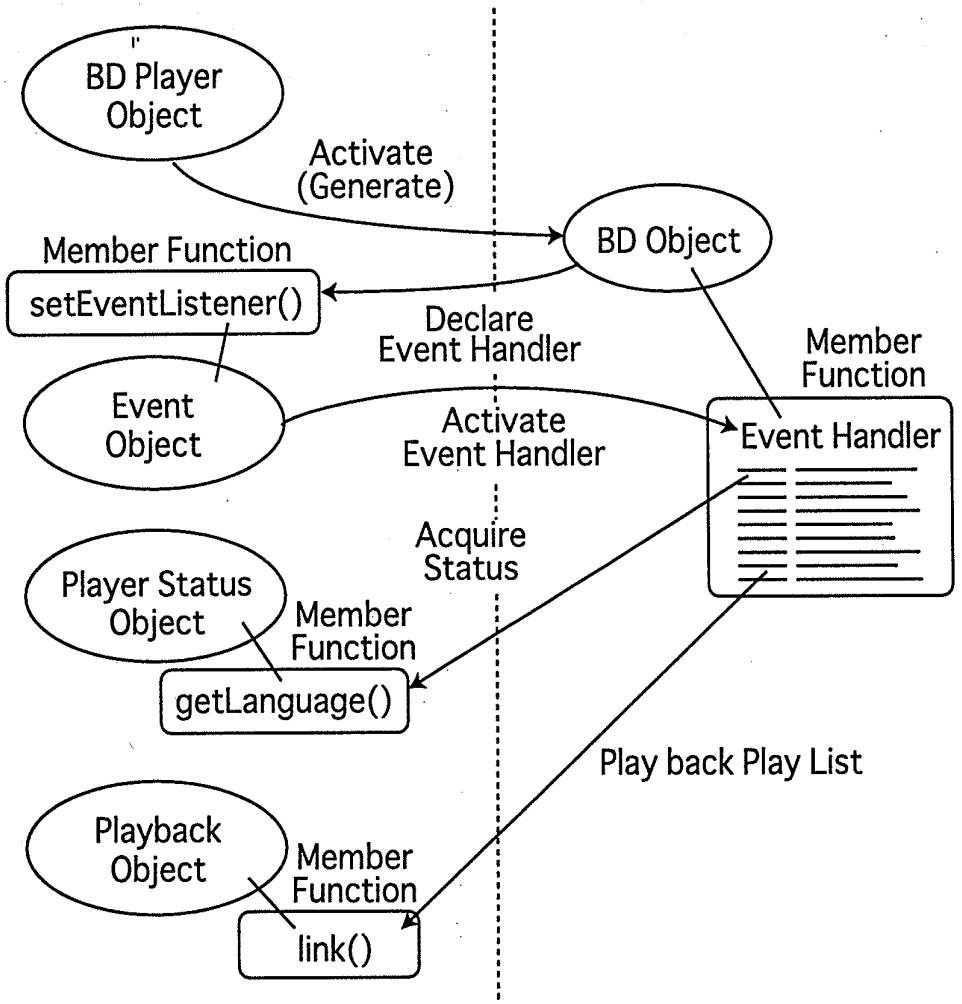

BD Player Object
Activate (Generate)
BD Object
Member Function
setEventListener()
Declare Event Handler
Member Function
Event Object
Activate Event Handler
Event Handler
Acquire Status
Player Status Object
Member Function
getLanguage()
Play back Play List
Playback Object
Member Function
link()

FIG.38

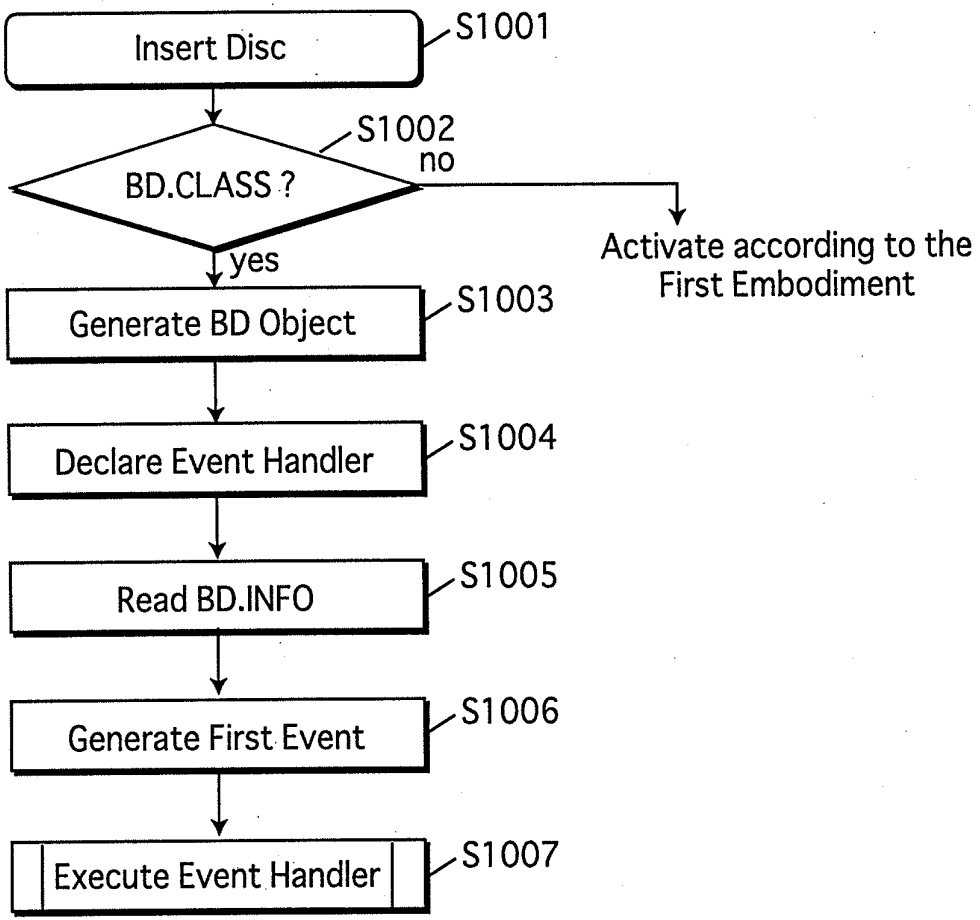

Insert Disc
S1001
S1002
BD.CLASS ?
no
yes
Activate according to the First Embodiment
Generate BD Object
S1003
Declare Event Handler
S1004
Read BD.INFO
S1005
Generate First Event
S1006
Execute Event Handler
S1007

FIG.39

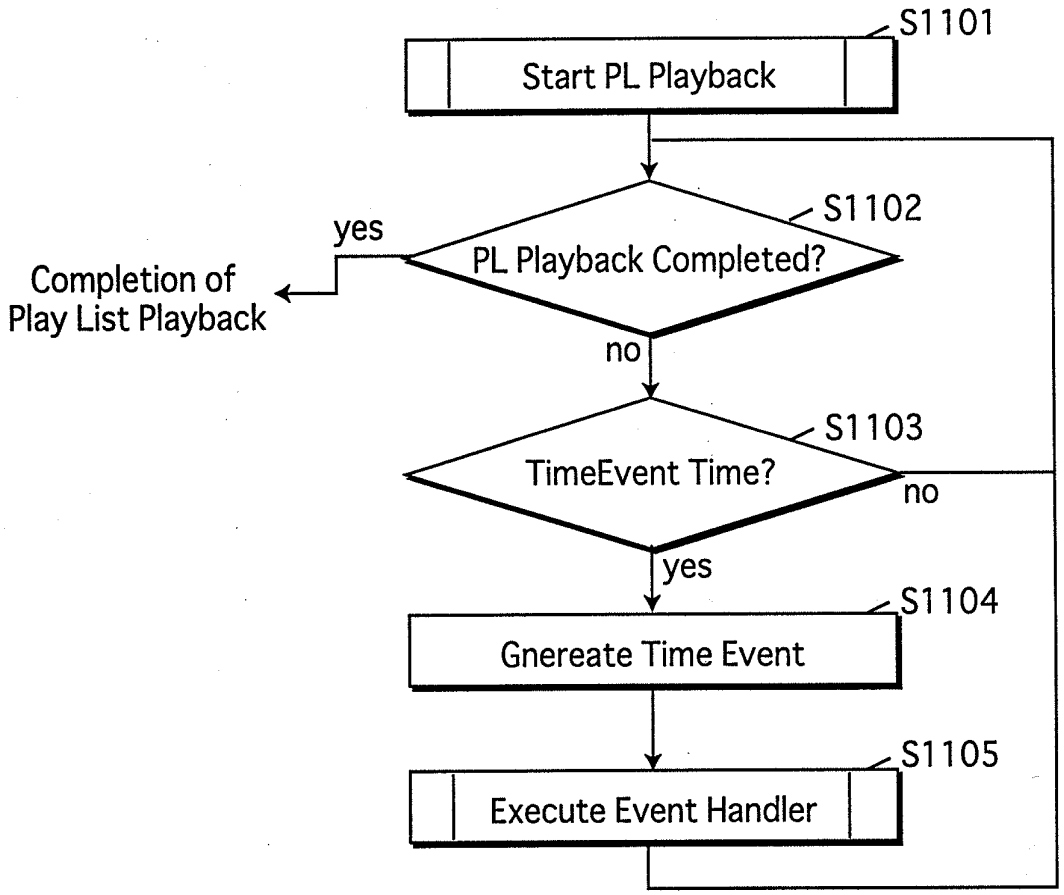
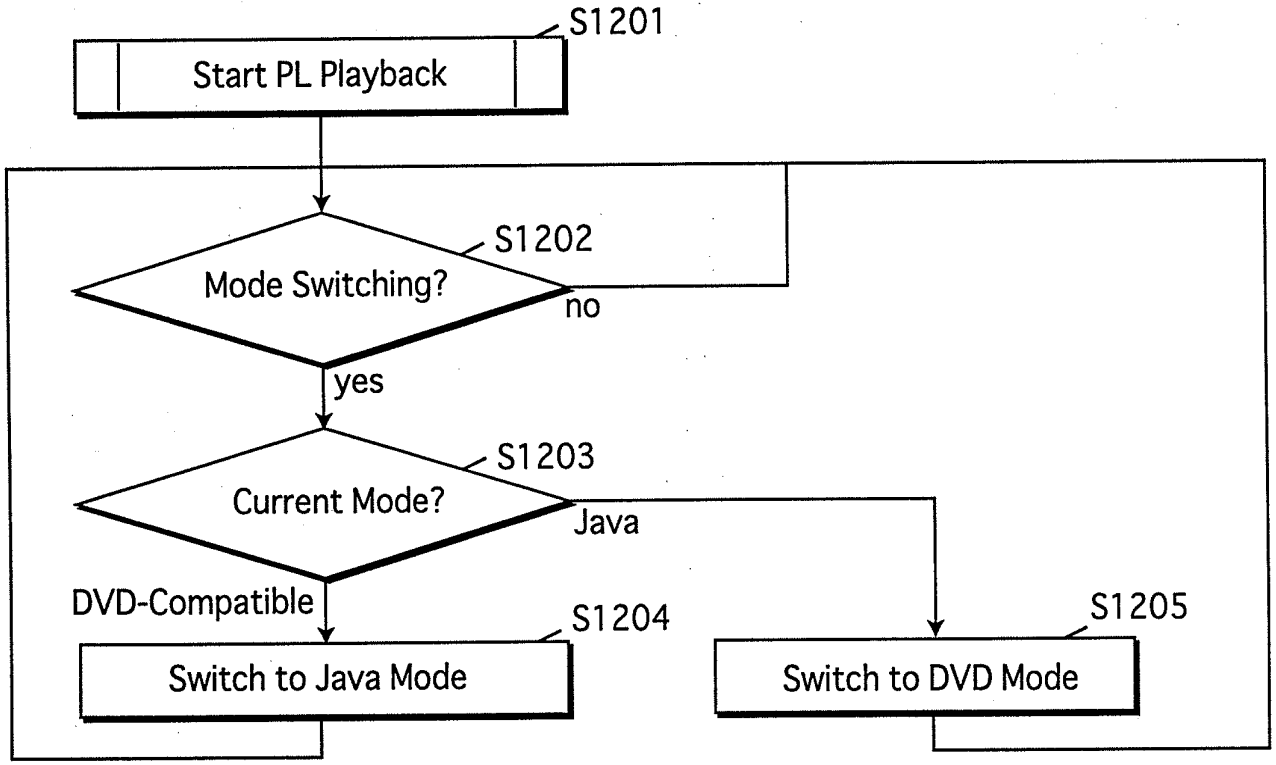

S1101
Start PL Playback
S1102
PL Playback Completed?
yes
Completion of
Play List Playback
no
S1103
TimeEvent Time?
no
yes
S1104
Gnereate Time Event
S1105
Execute Event Handler

FIG.40


S1201
Start PL Playback
S1202
Mode Switching?
no
yes
S1203
Current Mode?
Java
DVD-Compatible
S1204
Switch to Java Mode
S1205
Switch to DVD Mode